한국산학기술학회논문지
Vol. 8, No. 6, pp. 1480-1490, 2007

WINDOWS CE .NET 기반의 PDA를 이용한 원격제어시스템 개발

양원석[1], 이유상[1], 전재욱[1], 문일현[1], 최관순[1*], 전창완[1], 안달[1], 임종식[1]

A Development of Remote Control System using PDA based WINDOWS CE .NET

Wonseok Yang[1], Yoosang Lee[1], Jaewuk Jeon[1], Ilhyeon Moon[1], Kwan-Sun Choi[1*], ChangWan Jeon[1], Dal Ahn[1] and Jong-Sik Lim[1]

요 약 본 논문에서는 WINDOWS CE .NET 기반의 PDA를 이용한 원격제어시스템을 구현하였다. 서버 시스템의 운영체제는 WINDOWS CE .NET을 사용하여 사용자에게 쉬운 사용 환경을 제공하며, 운영체제 이미지에 개발한 디바이스 드라이버를 추가하여 하드웨어 장치를 제어한다. 임베디드 서버는 PC 클라이언트와 PC 클라이언트, PC 클라이언트와 PDA 클라이언트, PDA 클라이언트와 PDA 클라이언트 간의 메시지 통신을 지원한다.

본 논문은 원격제어시스템을 구현하는데 있어서 에뮬레이터를 사용하지 않고 실제 PDA와 임베디드 보드로 시스템을 구현하였다. 제안된 시스템은 임베디드 보드를 서버로 사용하고 PDA를 클라이언트로 사용하여 무선 인터넷이 제공되는 환경이라면 어느 곳에서나 원격으로 대상물을 제어할 수 있다.

Abstract This study implemented a remote control system using PDA based on WINDOWS CE. NET. In existing remote control systems, the user generally controls remote objects and obtain data using a computer, which is a local system connected to the Internet, and the remote objects are connected to server computers. In order to control remote objects, the user has to have a computer, a space to use the Internet, and wired Internet.

The system proposed in this study uses an embedded board as a server and PDA as a client. Thus, the user can control remote objects at any place as long as wireless Internet is available. In addition, because an embedded board is used as a server, the system has the characteristics of high performance and low power consumption. The proposed system provides not only remote control environment using PDA but also conventional control environment using a computer as a client. In case PDA is not available, remote objects still can be controlled from any place as long as there are wired Internet and a computer.

Key Words : Windows CE .NET, Remote Control, PDA, 원격제어

1. 서 론

최근 인터넷의 발달은 통신의 발전뿐만 아니라 일반 생활에 많은 변화를 가져왔다. 초기의 인터넷은 유선으로 연결되었지만 사용자가 증가하고, 장소의 제약에 따른 사용의 한계가 있어 무선으로 연결되는 인터넷이 증가하였고, 그 사용빈도가 점점 증가하는 추세이다. 이러한 인터넷의 발전은 독립적인 고유의 기능만을 수행하던 가정용, 산업용 기기들을 먼 거리에서 제어, 감시하는 원격제어시스템을 위한 많은 연구가 이루어지고 있다. 원격제어시스템은 공장 자동화, 군사 시스템, 우주 탐사뿐만 아니라 가정 자동화나 의료장비 등 다양한 분야에 쓰이고 있으며 관련연구가 활발히 진행 중이다.

인터넷 기반의 원격제어시스템은 일반적으로 사용자가 로컬 시스템인 컴퓨터에 연결되어 있는 인터넷을 사용하여 원격지에 있는 대상물을 제어하고, 데이터를 취득하는 시스템으로 구성되어있다. 원격제어시스템에서는 대상물을 제어하기 위해 컴퓨터와 인터넷을 사용해야만

본 연구는 산업자원부 지정 순천향대학교 차세대 BIT 무선부품지역혁신센터의 지원에 의한 것입니다.
[1]순천향대학교 전기통신공학과
*교신저자 : 최관순(cks1329@sch.ac.kr)

한다. 이것은 컴퓨터를 사용해야 할 장소와 유선으로 연결되는 인터넷이 있어야 하는 공간적 제약이 있다. 이러한 문제는 고집적화가 가능함에 따라 하나의 칩에 여러 가지 기능을 가지는 다기능 프로세서의 등장으로 인해 임베디드 시스템이 급격하게 발전함으로 해결되고 있다. 이러한 임베디드 시스템의 발전은 장소에 제약을 받지 않는 PDA와 같은 휴대용 컴퓨터의 사용을 증가시키고 있다. 이러한 PDA와 무선 인터넷의 발달로 어느 곳에서나 필요한 정보를 얻을 수 있게 되었다. 임베디드 시스템은 범용 컴퓨팅 시스템과 달리 자신을 포함하고 있는 기기에 부과된 특정 목적의 컴퓨팅 작업만을 수행한다. 일반적으로 임베디드 시스템은"특정 목적을 위하여 동작하는 컴퓨팅 시스템"이라고 정의할 수 있다. 따라서 범용 컴퓨터를 제외하고 컴퓨팅 시스템이 내장된 모든 시스템이 임베디드 시스템이다.

임베디드 시스템의 운영체제로는 크게 Windows CE와 임베디드 리눅스가 있다. Windows CE는 Windows와 유사한 사용자 환경을 제공하기 때문에 처음 접하는 사용자가 편리하게 사용할 수 있고, Windows 기반의 컴퓨터와 데이터 공유 및 교환이 쉬운 장점이 있다. 임베디드 리눅스는 Windows CE에 비해 커널이 가벼운 장점이 있지만 응용 프로그램의 개발과 Windows 기반의 컴퓨터와 데이터 공유, 교환이 어렵다는 단점이 있다.

본 논문에서는 PXA255 임베디드 보드와 PDA를 이용하여 Windows CE .NET 기반의 PDA형 원격제어시스템을 제안하였다. 제안한 시스템은 무선인터넷이 되는 곳이라면 어느 곳에서나 PDA를 이용하여 원격지에 있는 대상물 제어가 가능하다. 또한 제어해야 할 대상물을 임베디드 보드에 연결하여 공간 사용의 제한을 축소시키고, 고성능/저전력으로 제어해야 할 대상물을 제어함으로써 더 좋은 성능을 제공한다. 운영체제는 Windows CE .NET을 사용하여 사용자에게 좀 더 쉬운 제어 환경을 제공한다.

본 논문은 2 장 연구 동향에서 현재 임베디드 시스템의 연구 동향에 대해 살펴본다. 3장 설계 및 구현에서는 본 논문에서 구현한 원격제어시스템의 구성과 개발한 디바이스 드라이버, 임베디드 서버, PC 클라이언트, PDA 클라이언트에 대해 설명하였으며, 마지막으로 4장 결론에서는 원격제어시스템에 대한 구현 결과를 정리하고, 연구 개선방안을 제안하였다.

2. 임베디드 시스템 연구동향

임베디드 시스템은 특수 목적의 장치에 포함되어 핵심 기능을 수행하는 하드웨어 및 소프트웨어가 복합된 장치를 말한다. 임베디드 시스템 연구는 임베디드 시스템 하드웨어 연구와 임베디드 시스템 소프트웨어 연구로 분류된다.

2.1 임베디드 시스템 하드웨어 연구동향

전통적인 임베디드 시스템 하드웨어 연구는 개별 분야에 각각 특징적인 하드웨어를 설계 구현하였다. 하지만 최근에는 하나의 플랫폼으로 다양한 분야에 적용하려는 재활용성을 강조한 방향으로 연구가 진행되고 있다. 하드웨어를 구성하는 프로세서도 고성능화되고 있고 하드웨어 플랫폼은 일부 기능을 재설정할 수 있는 유연한 구조를 갖는 방향으로 발전하고 있다. 임베디드 시스템 하드웨어 연구에서 가장 핵심적인 기술 분야는 시스템온칩 분야이다. 다양한 기능을 하나의 칩으로 구성하는 시스템온칩 분야의 혁신적인 발전으로 임베디드 시스템의 경량화, 저전력화, 다기능화를 가능하게 하였다.[9]

2.2 임베디드 시스템 소프트웨어 연구동향

임베디드 소프트웨어 연구는 크게 임베디드 운영체제, 임베디드 미들웨어, 임베디드 소프트웨어 기본응용으로 분류된다. 국내의 경우 임베디드 운영체제 보다는 임베디드 미들웨어와 임베디드 응용 소프트웨어의 개발이 활발히 진행 중이다.

1) 임베디드 운영체제에 관한 연구

임베디드 운영체제는 저전력 지원과 소형 LCD에 적합한 그래픽 윈도우 시스템 기술이 중요하다. 또한 빠르게 변하는 시장에 대응하기 위해 임베디드 운영체제는 이식성과 조립성이 우수해야 한다. 임베디드 운영체제 연구는 작은 용량, 빠른 부팅, 저전력 지원, 다양한 프로세서 지원, 간결한 구조, 빠른 응답, 멀티 태스킹, 메모리 기반 파일 시스템, 실시간 지원을 위한 선점형 커널(preemptible kernel)지원, 락브레이크(lock break), 실시간 자원 스케쥴 등에 대한 연구가 활발히 진행 중이다[10].

2) 임베디드 미들웨어에 관한 연구

임베디드 미들웨어는 네트워크 상에서 제공되는 서비스와 이러한 서비스를 사용하는 소프트웨어 구성 요소들 간에 자연스러운 상호작용이 제공되도록 단순화된 하부 구조, 또는 소프트웨어 버스를 제공한다. 미들웨어의 이런 기능은 서비스들이 안정적으로 네트워크에 참가하거나 떠날 수 있도록 해주며, 클라이언트에게는 일람된 서비스의 가용성을 보장해준다.

HomePNA(Home Phoneline Networking Alliance [6])

와 전력선 네트워크, IEEE 1394 등의 다양한 네트워크 미디어와 정보가전 기기들로 구성된 홈 네트워크의 복잡성과 이질성은 분산 임베디드 시스템 프로그래머에게 중대한 설계 문제를 안겨주게 된다. 이런 문제들은 CORBA[7], DCOM[8], Jini[9], UPnP[10] 등과 같은 미들웨어 기술들을 통해서 부분적으로 해결이 될 수 있다. 최근 자원이 제한된 임베디드 시스템을 위해서 이러한 미들웨어들을 특화시키는 방법에 관한 활발한 연구가 진행되고 있다[11, 12, 13].

3) 임베디드 소프트웨어 기본응용에 관한 연구

임베디드 응용 및 서비스는 다양한 임베디드 시스템을 통해 사용자가 만날 수 있는 소프트웨어이다. 기본응용에는 브라우저와 미디어 플레이어 등이 있다. 응용 소프트웨어는 정보가전이 활성화됨에 따라 명령어 중심의 사용자 환경에서 그래픽 중심의 사용자 환경으로 변화되고 있다. 키보드와 마우스 등에서 벗어나 자연스럽게 컴퓨터와 소통하는 사용자 환경에 대해 연구가 활발히 진행되고 있다.[14, 15]

3. 원격제어시스템 설계 및 구현

3.1 원격제어시스템 구성

제안한 원격제어시스템은 크게 임베디드 서버, PDA 클라이언트, 컴퓨터 클라이언트 시스템으로 구성된다. 임베디드 서버는 PDA 클라이언트 및 컴퓨터 클라이언트와 연결된다. 임베디드 서버에 제어할 하드웨어를 장착하고 디바이스 드라이버를 개발하여 적재한다. 각 클라이언트는 원격에서 임베디드 시스템에 연결되어 있는 하드웨어를 제어할 수 있다. <그림 1>은 원격제어 시스템의 구성을 보여준다.

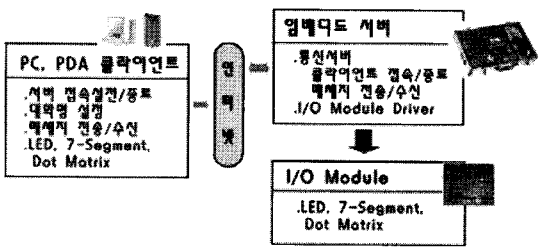


그림 1. 원격제어시스템의 구성도

3.2 개발 환경

1) 하드웨어 환경

표1은 하드웨어 개발환경으로, 원격제어시스템은 디바이스 드라이버, 임베디드 서버 프로그램, 컴퓨터 클라이언트 프로그램을 개발하기 위한 컴퓨터와 PDA 클라이언트 프로그램을 개발하기 위한 컴퓨터, 2대의 컴퓨터를 사용하였다. 또 서버 프로그램이 동작할 임베디드 보드 1대, PDA 클라이언트 프로그램이 동작할 PDA 1대를 사용하였다. 총 2대의 컴퓨터와 임베디드 보드, PDA를 사용하였다.

표 1. 개발 하드웨어 환경

항목	CPU	RAM	통신포트
서버 개발 환경	P4 3.2GHz	1Gbyte	100Mbps 이더넷, 시리얼포트, 프린터포트, USB포트
PDA 클라이언트개발 환경	P4 3.0GHz	1Gbyte	100Mbps 이더넷, 시리얼포트, 프린터포트, USB포트
임베디드 보드	Intel Xscale PXA255 400MHz	64Mbyte SDRAM	100Mbps 이더넷,시리얼포트, JTAG, USB
PDA	Intel Xscale PXA270 520MHz	256Mbyte	블루투스, Wi-Fi 무선랜, USB

표 2. 개발 소프트웨어 환경

항목	운영체제	개발도구	통신도구
임베디드 서버 개발 환경	Windows XP	eMbedded Visual C++ 4.0, Visual C++ 6.0, Platform Builder 4.2	JFlashMM 5.1.2600, ActiveSync 4.2.0
PDA 클라이언트 개발 환경	Windows XP	Visual Studio 2005	ActiveSync 4.2.0
임베디드 보드	Windows CE .NET 4.2		
PDA	Microsoft® Windows Mobile™ 5.0 포켓 PC		

2) 소프트웨어 환경

표2는 개발 소프트웨어 환경으로 임베디드 서버 프로그램 개발은 eVC를 사용하였고 PDA 클라이언트 프로그

램을 개발하기 위해 Visual Studio 2005를 사용하였다. 또한 컴퓨터 클라이언트 프로그램을 개발하기 위해 Visual Studio 6.0을 사용하였다.

3.3 디바이스 드라이버

본 논문에서는 스트림 디바이스 드라이버를 이용하여 LED, 7-세그먼트, 도트 매트릭스의 디바이스 드라이버를 구현하였다. 디바이스 드라이버를 개발하기 위해서는 우선 HAL 소스를 편집하여 하드웨어의 물리 주소와 Windows CE에서 사용하는 가상 주소를 연결하여야 한다. 그 후 각 장치별로 def 파일과 sources 파일을 만든다. def 파일은 해당 디바이스 드라이버가 사용하는 함수를 정의하고, sources 파일은 생성될 드라이버의 빌드와 관련된 내용을 정의한다. 구현된 디바이스 드라이버는 DLL 파일의 형태로 Windows CE .NET 운영체제 이미지에 포함되어 동작한다. 표3은 LED, 7-세그먼트, 도트매트릭스의 디바이스 드라이버 def파일을 나타낸다.

표 3. 디바이스 드라이버 파일

드라이버	def 파일
LED	LIBRARY LED_DEV EXPORTS LEX_Init LEX_Deinit LEX_Open LEX_Close LEX_Read LEX_Write LEX_Seek LEX_IOControl LEX_PowerDown LEX_PowerUp
7 세그먼트	LIBRARY FNX_DEV EXPORTS FNX_Init FNX_Deinit FNX_Open FNX_Close FNX_Read FNX_Write FNX_Seek FNX_IOControl FNX_PowerDown FNX_PowerUp FNX_DllEntry
도트 매트릭스	LIBRARY DOX_DEV EXPORTS DOX_Init DOX_Deinit DOX_Open DOX_Close DOX_Read DOX_Write DOX_Seek DOX_IOControl DOX_PowerDown DOX_PowerUp DOX_DllEntry

다음 코드는 LED 디바이스 드라이버의 sources 파일로 생성될 디바이스 드라이버의 정보를 가지고 있다.

```
TARGETNAME=led_dev
RELEASETYPE=PLATFORM
TARGETTYPE=DYNLINK

TARGETLIBS=$(_COMMONSDKROOT)\lib\$(_CPUINDPATH)\coredll.lib
SOURCELIBS=$(_TARGETPLATROOT)\lib\$(_CPUINDPATH)\drvlib.lib

DEFFILE=led_dev.def
PREPROCESSDEFFILE=1
DLLENTRY=DllEntry
INCLUDES=..\..\inc
SOURCES=led_dev.cpp
```

다음 소스는 7-세그먼트 디바이스 드라이버의 sources 파일로 생성될 디바이스 드라이버의 정보를 가지고 있다.

```
TARGETNAME=fnd_dev
RELEASETYPE=PLATFORM
TARGETTYPE=DYNLINK

TARGETLIBS=$(_COMMONSDKROOT)\lib\$(_CPUINDPATH)\coredll.lib
SOURCELIBS=$(_TARGETPLATROOT)\lib\$(_CPUINDPATH)\drvlib.lib

DEFFILE=fnd_dev.def
PREPROCESSDEFFILE=1
DLLENTRY=FNX_DllEntry
INCLUDES=..\..\inc;
SOURCES=fnd_dev.cpp
```

생성될 디바이스 드라이버의 정보를 갖는 도트 매트릭스 디바이스 드라이버의 소스파일은 다음과 같다.

```
TARGETNAME=dot_dev
RELEASETYPE=PLATFORM
TARGETTYPE=DYNLINK

TARGETLIBS=$(_COMMONSDKROOT)\lib\$(_CPUINDPATH)\coredll.lib
SOURCELIBS=$(_TARGETPLATROOT)\lib\$(_CPUINDPATH)\drvlib.lib

DEFFILE=dot_dev.def
PREPROCESSDEFFILE=1
DLLENTRY=DOX_DllEntry
INCLUDES=..\..\inc;
SOURCES=dot_dev.cpp
```

platform.bib 파일은 Windows CE .NET 운영체제 이미지 생성시 추가될 DLL들을 정의한다. platform.bib 파일에 LED 디바이스 드라이버를 추가한 부분이다.

```
led_dev.dll      $(_FLATRELEASEDIR)\led_dev.dll
                   NK    SH
```

platform.bib 파일에 7-세그먼트 디바이스 드라이버를 추가한 코드이다.

```
fnd_dev.dll      $(_FLATRELEASEDIR)\fnd_dev.dll
                   NK    SH
```

다음은 도트 매트릭스 디바이스 드라이버를 추가한 부분이다.

```
dot_dev.dll      $(_FLATRELEASEDIR)\dot_dev.dll
                 NK   SH
```

마지막으로 platform.reg는 Windows CE .NET 운영체제의 레지스트리로서 운영체제의 시스템 정보를 정의한다. 이 platform.reg에 LED 디바이스 드라이버를 추가한 코드이다.

```
[HKEY_LOCAL_MACHINE\Drivers\BuiltIn\LED_DEV]
   "Dll"="led_dev.dll"
   "Prefix"="LEX"
   "Index"=dword:1
   "Order"=dword:0
   "Ioctl"=dword:4
```

platform.reg에 7-세그먼트 디바이스 드라이버를 추가한 부분은 다음과 같다.

```
[HKEY_LOCAL_MACHINE\Drivers\BuiltIn\FND_DEV]
   "Dll"="fnd_dev.dll"
   "Prefix"="FNX"
   "Index"=dword:1
   "Order"=dword:0
   "Ioctl"=dword:4
```

platform.reg에 도트 매트릭스 디바이스 드라이버를 추가한 코드는 다음과 같다.

```
[HKEY_LOCAL_MACHINE\Drivers\BuiltIn\DOT_DEV]
   "Dll"="dot_dev.dll"
   "Prefix"="DOX"
   "Index"=dword:1
   "Order"=dword:0
   "Ioctl"=dword:4
```

3.4 임베디드 서버

임베디드 서버의 네트워크 프로그래밍 인터페이스는 윈도우 소켓을 사용하였다. 임베디드 서버는 클라이언트들을 연결하고, 클라이언트로부터 받은 메시지를 해석하여 하드웨어를 제어하거나 메시지를 다른 클라이언트로 전송한다. 임베디드 서버는 1개의 에디트 박스와, 1개의 버튼으로 구성되어 있다. 에디트 박스는 받은 메시지를 보여주고 서버의 상태를 표시하고, CLOSE 버튼은 서버를 종료시키는 기능을 한다.

1) 통신 서버

임베디드 서버는 클라이언트의 연결 접속 요청이 있을 때 클라이언트와 연결하여 주고, 수신된 메시지를 다른 클라이언트에게 전달하는 통신 서버로서의 기능을 한다.

가. 클라이언트 접속/ 종료

클라이언트가 접속하면 해당 클라이언트의 정보를 표시한다. 클라이언트가 종료되면 종료된 클라이언트의 IP 주소를 표시해주고 소켓을 닫는다. 클라이언트가 종료되어도 서버는 종료되지 않고 다른 클라이언트의 접속 요청을 대기해야 한다.

나. 메시지 송/수신

임베디드 서버는 클라이언트로부터 받은 메시지를 다른 클라이언트에게 전달해준다. 임베디드 서버와 PDA 클라이언트는 유니코드를 사용하고, PC 클라이언트는 아스키코드를 사용한다. 임베디드 서버는 모든 클라이언트로부터 메시지를 받아서 유니코드로 변환하여 사용하고, 다른 클라이언트에게 메시지를 전달할 때는 받은 아스키코드를 그대로 전송한다. PDA 클라이언트는 메시지를 전송하기 전에 아스키코드에서 유니코드로 메시지를 변환하여 전송하고, 서버로부터 아스키코드 메시지를 수신하여 유니코드로 변환해서 사용한다. 그림 2는 서버가 클라이언트로부터 메시지를 수신한 화면이다. 서버 실행 후 PDA와 PC 클라이언트가 접속하여 "가나다ABC123" 메시지를 수신한다.

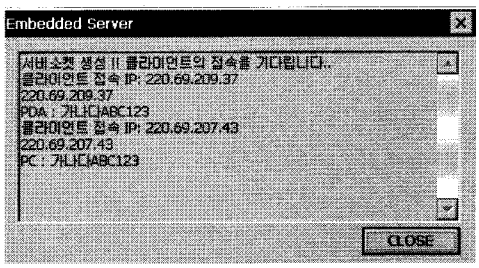


그림 2. 클라이언트에서 메시지 수신

2) 하드웨어 제어

가. LED 제어

LED 제어 메시지는 1자리의 장치 번호와 8자리의 데이터 메시지로 구성된다. LED는 장치 번호를 1로 설정하였으며 데이터 메시지는 제어하고자 하는 LED자리에 1을 입력하면 제어할 수 있다. 그림 19는 LED 제어 메시지로 LED를 제어한 결과이다.

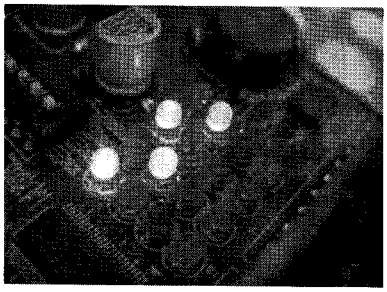


그림 3. LED 결과

나. 7-세그먼트 제어

7-세그먼트 제어 메시지는 1자리의 장치 번호와 6자리의 데이터 메시지로 구성된다. 7-세그먼트는 장치 번호를 2로 설정하였으며 데이터 메시지는 제어 하고자 하는 7-세그먼트의 자리에 숫자를 입력하면 제어할 수 있다. 그림 4는 7-세그먼트 제어 메시지로 7-세그먼트를 제어한 결과이다.

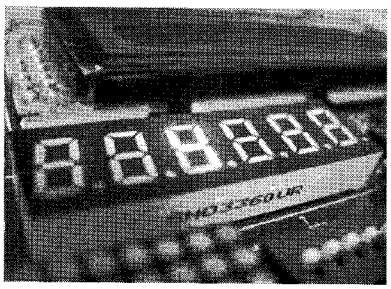


그림 4. 7-세그먼트 결과

다. 도트 매트릭스 제어

도트 매트릭스 제어 메시지는 1자리의 장치 번호와 1자리의 데이터 메시지로 구성된다. 도트 매트릭스의 장치 번호는 3으로 설정하였으며, 데이터 메시지에 숫자를 입력하면 도트 매트릭스에서 결과를 볼 수 있다. 그림 5는 도트 매트릭스 제어 메시지로 도트 매트릭스를 제어한 결과이다.

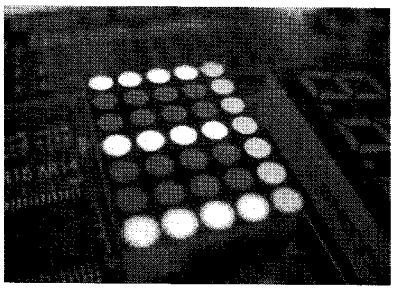


그림 5. 도트 매트릭스 결과

3.5 PC 클라이언트

PC 클라이언트는 컴퓨터에서 임베디드 서버에 접속한다. PC 클라이언트는 임베디드 서버를 통하여 다른 클라이언트와 메시지 통신을 하거나 하드웨어 제어 메시지를 전송하여 임베디드 서버에 있는 하드웨어를 제어한다. PC 클라이언트는 보내기, 접속, 해제, 대화명, 닫기, LED, SEGMENT, DOTMATRIX의 버튼으로 구성되어 있다. 각 버튼은 메시지 전송, 접속할 서버의 IP주소와 포트번호 설정, 접속해제, 대화명 설정, 프로그램종료, LED 제어, 7-세그먼트 제어, 도트 매트릭스 제어의 기능을 한다.

1) 서버 접속 설정/ 접속 종료

PC 클라이언트 실행후 접속 버튼을 누르면 그림 6의 접속 설정 대화상자가 실행된다. 접속할 서버의 IP주소와 포트번호를 입력하고 OK 버튼을 누르면 해당 IP 주소와 포트번호에 접근하여 접속 요청을 한다.

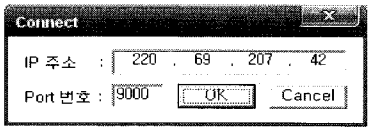


그림 6. 접속 설정

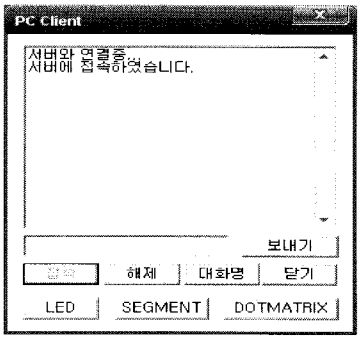


그림 7. 서버 접속

그림 8. 접속 종료

그림 7은 서버로부터 접속 허가 메시지를 받아 PC 클라이언트가 서버에 접속된 상태이다. PC 클라이언트가 서버에 접속하면 보내기, LED, SEGMENT, DOTMATRIX등 비활성화 되어있던 버튼들이 활성화 된다. 해제 버튼을 누르면 서버와의 연결이 종료되고 보내기, 해제, LED SEGMENT, DOTMATRIX 버튼은 비활성화 된다. 그림 8은 접속 종료 화면이다. 서버가 종료되어도 해제 버튼을 눌렀을 때와 같은 동작을 한다.

2) 대화명 설정

PC 클라이언트 프로그램에서 대화명 버튼을 누르면 그림 9의 대화명 설정 대화상자가 실행된다. 대화명은 서버에 접속하기 전이나 서버에 접속 후 언제든지 변경할 수 있다.

3) 메시지 전송/수신

PC 클라이언트가 서버에 접속되면 보내기 버튼이 활성화되면서 메시지를 전송할 수 있다. 전송할 메시지는 한글, 영문, 숫자를 조합하여 "Hello PDA! 가나다 123"을 작성하였다. 메시지 작성후 보내기 버튼을 누르면 서버로 메시지가 전송된다.

그림 10은 메시지를 전송한 화면이다. 메시지를 전송하면 서버에 전달되고 서버는 받은 메시지를 접속되어 있는 다른 클라이언트들에게 전달한다. 그림 11은 서버로부터 다른 클라이언트가 보낸 메시지를 수신한 화면이다.

그림 9. ID 설정

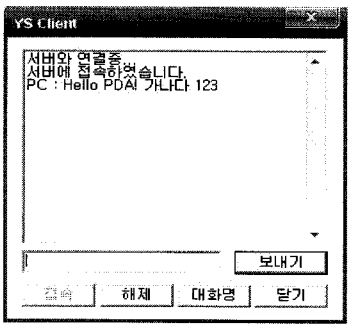


그림 10. 메시지 전송

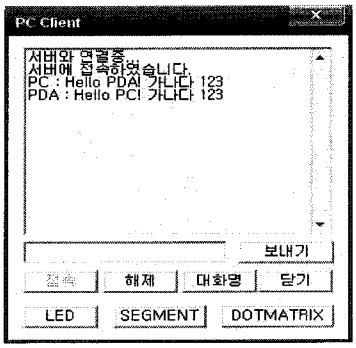


그림 11. 메시지 수신

4) LED / 7-세그먼트 / 도트 매트릭스 제어

PC 클라이언트 프로그램에서 LED 버튼을 누르면 그림 12의 LED 제어 대화상자가 실행된다. LED 제어 대화상자는 10개의 버튼으로 구성되어 있다. 8개의 버튼은 LED 제어 메시지를 설정하고 Send 버튼은 서버로 제어 메시지 전송, Close 버튼은 LED 제어 대화상자를 종료한다.

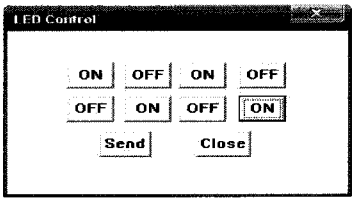


그림 12. LED 제어

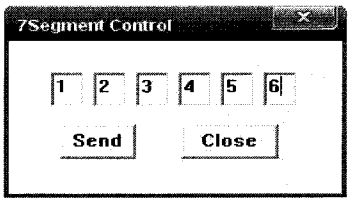


그림 13. 7-세그먼트 제어

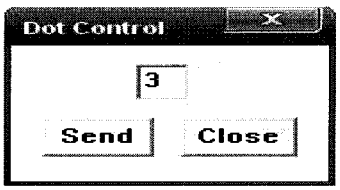


그림 14. 도트 매트릭스 제어

그림 12는 LED 제어 메시지를 작성하는 화면이다. OFF 상태의 버튼을 누르면 ON으로 바뀌고, ON 상태의 버튼을 누르면 OFF 상태로 바뀐다. PC 클라이언트 프로

그램에서 SEGMENT 버튼을 누르면 그림 13의 7-세그먼트 제어 대화상자가 실행된다. 7-세그먼트 제어 대화상자는 6개의 에디트 박스와 2개의 버튼으로 구성되어 있다.

3.6 PDA 클라이언트

PDA 클라이언트는 PDA에서 임베디드 서버에 접속한다. PDA 클라이언트는 임베디드 서버를 통하여 다른 클라이언트와 메시지 통신을 하거나 하드웨어 제어 메시지를 전송하여 임베디드 서버에 있는 하드웨어를 제어한다. PDA 클라이언트는 Send, Connect, Disconnect, ID, Close, LED, 7-SEGMENT, DOTMATRIX의 버튼으로 구성되어 있다. 각 버튼은 메시지 전송, 접속할 서버의 IP주소와 포트번호 설정, 접속해제, 대화명 설정, 프로그램종료, LED 제어, 7-세그먼트 제어, 도트 매트릭스 제어의 기능을 한다.

1) 서버 접속 설정/ 접속 종료

PDA 클라이언트 실행후 Connect 버튼을 누르면 그림 15의 접속 설정 대화상자가 실행된다. 접속할 서버의 IP 주소와 포트번호를 입력하고 OK 버튼을 누르면 해당 IP 주소와 포트번호에 접근하여 접속 요청을 한다.

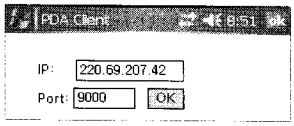


그림 15. 접속 설정

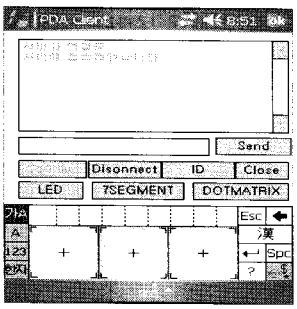


그림 16. 서버 접속

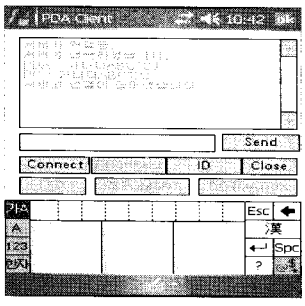


그림 17. 접속 종료

Disconnect 버튼을 누르면 서버와의 연결이 종료되고 Send, Disconnect, LED, 7SEGMENT, DOTMATRIX 버튼은 비활성화 된다. 그림 16은 서버로부터 접속 허가 메시지를 받아 PDA 클라이언트가 서버에 접속된 상태이다. PDA 클라이언트가 서버에 접속하면 Send, LED, 7SEGMENT, DOTMATRIX 등 비활성화 되어 있던 버튼들이 활성화 된다. 그림 17은 PDA 클라이언트의 접속 종료 화면이다. 서버가 종료되어도 해제 버튼을 눌렀을 때와 같은 동작을 한다.

2) 대화명 설정

PDA 클라이언트 프로그램에서 ID 버튼을 누르면 그림 18의 대화명 설정 대화상자가 실행된다. 대화명은 서버에 접속하기 전이나 서버에 접속 후 언제든지 변경할 수 있다.

3) 메시지 전송/ 메시지 수신

PDA 클라이언트가 서버에 접속되면 Send 버튼이 활성화되면서 메시지를 전송할 수 있다. 전송할 메시지는 한글, 영문, 숫자를 조합하여 "Hello PC! 가나다 123"을 작성하였다. 메시지 작성 후 Send 버튼을 누르면 서버로 메시지가 전송된다.

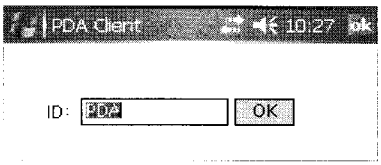


그림 18. ID 설정

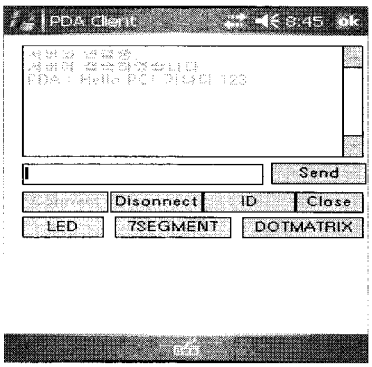


그림 19. 메시지 전송

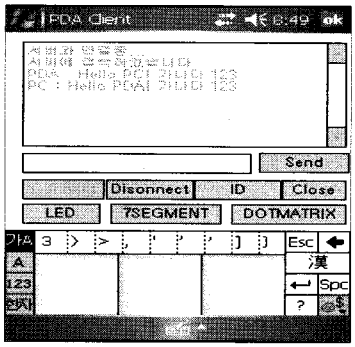


그림 20. 메시지 수신

그림 19는 메시지를 전송한 화면이다. 메시지를 전송하면 서버에 전달되고 서버는 받은 메시지를 접속되어 있는 다른 클라이언트들에게 전달한다. 그림 20은 서버로부터 다른 클라이언트가 보낸 메시지를 수신한 화면이다.

4) LED / 7-세그먼트 / 도트 매트릭스 제어

PDA 클라이언트 프로그램에서 LED 버튼을 누르면 그림 21의 LED 제어 대화상자가 실행된다. LED 제어 대화상자는 10개의 버튼으로 구성되어 있다. 8개의 버튼은 LED 제어 메시지를 설정하고 Send 버튼은 서버로 제어 메시지 전송, Close 버튼은 LED 제어 대화상자를 종료한다.

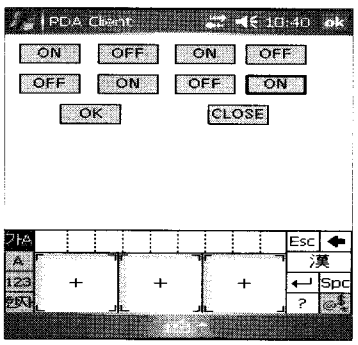


그림 21. LED 제어

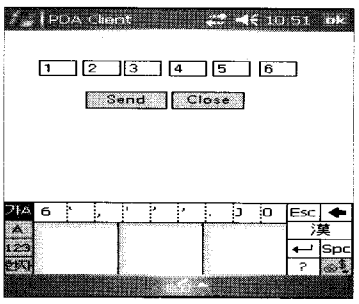


그림 22. 7-세그먼트 제어

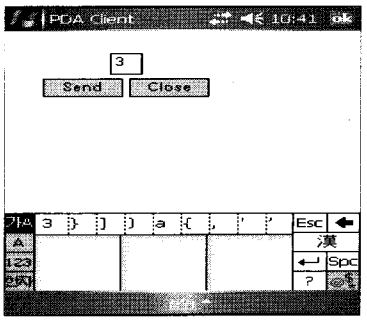


그림 23. 도트 매트릭스 제어

그림 22는 LED 제어 메시지를 작성하는 화면이다. OFF 상태의 버튼을 누르면 ON으로 바뀌고, ON 상태의 버튼을 누르면 OFF 상태로 바뀐다. PDA 클라이언트 프로그램에서 7SEGMENT 버튼을 누르면 그림 23의 7-세그먼트 제어 대화상자가 실행된다. 7-세그먼트 제어 대화상자는 6개의 에디트 박스와 2개의 버튼으로 구성되어 있다. 6개의 에디트 박스는 7-세그먼트 제어 메시지를 설정하고, Send 버튼은 서버로 제어 메시지 전송, Close 버튼은 7-세그먼트 제어 대화상자를 종료한다. 그림 23은 7-세그먼트 제어 메시지를 작성하는 화면이다. 각 에디트 박스에 숫자를 입력하고 메시지를 전송하면 서버 측에 있는 임베디드 장비의 7-세그먼트의 자리에 보낸 메시지가 출력된다. PDA 클라이언트 프로그램에서 DOTMATRIX 버튼을 누르면 그림 23의 도트 매트릭스 제어 대화상자가 실행된다. 도트 매트릭스 제어 대화상자는 1개의 에디트 박스와 2개의 버튼으로 구성되어 있다. 에디트 박스는 도트 매트릭스 제어 메시지를 설정하고, Send 버튼은 서버로 제어 메시지 전송, Close 버튼은 도트 매트릭스 제어 대화상자를 종료한다. 그림 23은 도트 매트릭스 제어 메시지를 작성하는 화면이다. 에디트 박스에 숫자를 입력하고 메시지를 전송하면 서버 측에 있는 임베디드 시스템의 도트 매트릭스에 숫자 모양으로 출력된다.

5. 결 론

본 논문은 Windows CE .NET을 기반으로 하여 PDA 타입의 원격제어시스템을 설계 및 구현함으로써 무선인터넷 환경이 제공되는 곳이라면 시간과 공간의 제약 없이 원격 제어가 가능한 시스템을 제안하였다. 뿐만 아니라 무선인터넷 환경이 제공되지 않더라도 유선인터넷 환경이 제공되고 컴퓨터만 있으면 원격 제어가 가능하다.

컴퓨터를 서버 시스템으로 사용하여 개발된 원격제어 시스템의 경우 모니터, 본체, 키보드, 마우스 등의 주변장치로 인하여 적지 않은 공간을 차지하고 비용 또한 적지 않다. 본 논문에서는 임베디드 보드를 서버 시스템으로 사용하여 이러한 문제를 해결하였다.

서버 시스템의 운영체제는 Windows CE .NET을 사용하여 사용자에게 쉬운 사용 환경을 제공하였다. 운영체제 이미지에 본 연구에서 개발한 디바이스 드라이버를 추가하여 하드웨어 장치를 제어하도록 하였다.

본 연구에서는 클라이언트는 PDA와 PC를 모두 사용하였다. PC 클라이언트로 임베디드 서버에 접속하여 하드웨어 장치를 제어할 수 있었으며, 또한 PDA 클라이언트도 임베디드 서버에 접속하여 하드웨어 장치를 제어할 수 있게 만들었다. 그리고 본 연구에서는 임베디드 서버를 통해 PC 클라이언트와 PC 클라이언트, PC 클라이언트와 PDA 클라이언트, PDA 클라이언트와 PDA 클라이언트 간의 메시지 통신을 지원하게 하였다.

본 논문은 원격제어시스템을 구현하는데 있어서 에뮬레이터를 사용하지 않고 실제의 PDA와 임베디드 보드로 시스템을 구현하였다. 본 논문에서는 LED, 7-세그먼트, 도트 매트릭스 등 간단한 소자를 제어하여 원격 제어의 가능성만을 제안하였으며, 이를 이용하여 가정자동화나 공장자동화 시스템을 직접 구축할 수 있을 것이다.

참고문헌

[1] S. Hong. Embedded Linux Outlook in the PostPC Industry, In IEEE International Symposium on Object-oriented Real-time distributed Computing. pp. 37-40, May(.) 2003.

[2] 권문영(2006). PDA를 활용한 모바일 영어마을 모형 연구. 석사학위 논문, 중앙대학교.

[3] 김현규(2005). 임베디드 시스템 활용 교육을 위한 가상교육 시스템. 석사학위 논문, 순천향대학교.

[4] 박은경(2005). U-healthcare를 위한 ARM 프로세서와 실시간 운영체제 기반의 임베디드 시스템 구현. 석사학위 논문, 한양대학교.

[5] 서석현(2006). 차량내 네트워크를 위한 게이트웨이 임베디드 시스템. 석사학위 논문, 성균관대학교.

[6] 성하준(2006). 지능형 홈 서버를 위한 임베디드 리눅스 기반의 플랫폼 개발. 석사학위 논문, 전남대학교.

[7] 엄윤기(2004). ARM 기반의 JPEG2000 임베디드 시스템 구현. 석사학위 논문, 경북대학교.

[8] 유용덕(2006). 임베디드 디바이스를 위한 컴포넌트 기반 경량미들웨어. 석사학위 논문, 충남대학교.

[9] 전홍구(2002). 웹 기반 하드웨어 제어기술을 이용한 실시간 원격 실험 시스템의 설계 및 구현. 박사학위 논문, 순천향대학교.

[10] 김재명, 박태준, 양만석, 권기구, 임동선(2006). 차세대 임베디드 시스템을 위한 소프트웨어 플랫폼 현황 및 동향. 전자동신동향분석. 21권 1호. 57-67.

[11] 김홍남(2004). 임베디드 S/W 최신 기술 동향. 전자공학회지. 31권 11호. 19-29.

[12] 이형석, 정영준(2006). 임베디드 운영체제 커널 기술 동향. 전자통신동향분석. 21권 1호. 1-13.

[13] 임베디드 시스템 개론. 서울 : (주)휴인스 기술연구소.

[14] Windows CE .NET 응용과 실습. 서울 : (주)휴인스 기술연구소.

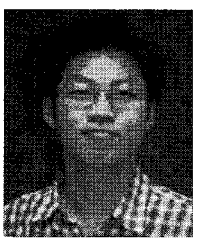
[15] Object Management Group. The Common Object Request Broker: Architecture and specification revision 2.2, Feb. 1998.

이 유 상(Yoosang Lee) **[준회원]**

- 2006년 2월 순천향대학교 정보기술공학부 졸업
- 2007년 현재 순천향대학교 전기통신시스템공학과 석사과정

<관심분야>
원격하드웨어제어, 임베디드시스템, 원격교육, 영상처리

문 일 현(ILHyeon Moon) **[준회원]**



- 2007년 2월 순천향대학교 정보기술공학부 졸업
- 2007년 현재 순천향대학교 전기통신시스템공학과 석사과정

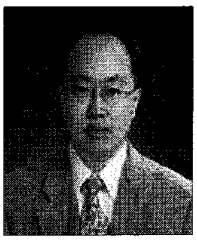
<관심분야>
임베디드시스템, 원격교육, 영상처리

전 재 욱(Jaewuk Jeon) [준회원]

- 2006년 2월 순천향대학교 정보기술공학부 졸업
- 2007년 현재 순천향대학교 전기통신시스템공학과 석사과정

<관심분야>
RFID, 임베디드시스템, 가상 교육 콘텐츠개발, 영상처리, 컴퓨터그래픽스

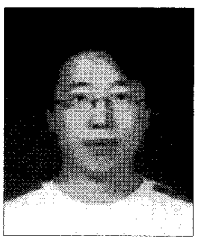
전 창 완(ChangWan Jeon) [정회원]



- 1996년 : 서울대학교 공과대학 제어계측공학과 (공학박사)
- 2006 현재 : 순천향대학교 전기통신시스템공학과 부교수

<관심분야>
제어시스템, 시스템모델링, GPS응용시스템, 임베디드시스템SW, 응용 SW

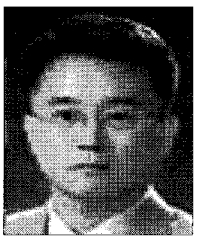
양 원 석(Wonseok Yang) [준회원]



- 2005년 8월 순천향대학교 정보기술공학부 졸업
- 2007년 현재 순천향대학교 전기통신시스템공학과 석사과정

<관심분야>
임베디드시스템, 가상 교육 콘텐츠개발, 영상처리

안 달(Dal Ahn) [정회원]



- 1990년 서강대학교 전자공학과 박사
- 2005년 이후 순천향대학교 전기통신시스템공학과 교수

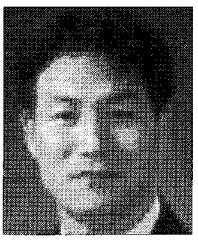
<관심분야>
마이크로웨이브 수동소자, 각종 FIlter, 부품 및 마이크로웨이브 회로설계

최 관 순(Kwansun Choi) [정회원]

- 1994년 8월 : 서강대학교 전자공학과(공학박사)
- 2006 현재 : 순천향대학교 전기통신시스템공학과 교수

<관심분야>
영상처리, GIS, 가상교육, 임베디드시스템, RFID, 웹기반 하드웨어제어, 영상처리, 교육용콘텐츠개발

임 종 식(Jongsik Lim) [정회원]



- 1993년 2월 : 서강대학교 전자공학과 초고주파 전공(공학석사)
- 2003년 2월 : 서울대학교 전기컴퓨터공학부(공학박사)
- 2005년 3월~ 현재: 순천향대학교 전기통신시스템공학과 교수

<관심분야>
초고주파 능동/수동회로 설계, 주기구조의 모델링 및 회로 응용, 안테나 측정 시스템 등