

## 무선 임베디드 주문 시스템 Wireless Ordering System For Embedded

김민성, 조상희, 정성철, 이상규, 홍경호  
백석대학교

Kim min-seong, Cho sang-hee, Jeong seong-cheol,  
Lee sang-gyu, Hong kyung-ho  
Baekseok Univ.

### 요약

본 논문에서는 Qt 임베디드를 이용한 무선 주문 시스템을 구성함으로써 실용성 있는 제품 제작에 활용될 수 있는 기법을 제안하고, 실제 구현하여 다양한 응용에 활용할 수 있음을 알 수 있다. 본 논문에서는 무선 네트워크 상에서 Qt에서 제공되는 QSocket 클래스의 TCP/IP 프로토콜을 이용하여 데이터 입력, 전달, 수집, 재전송 등을 구현하였다. 서버 환경은 PC 상에서 Linux 9.0, gcc-2.95.3을 사용하였으며, PXA255 프로세서를 사용하는 3 대의 타겟 보드를 모두 액세스 포인트를 기반으로 하는 무선 네트워크로 구현하였다. 타겟 보드의 소프트웨어 툴은 gcc, tmake, qt-X11, qt-embedded를 사용하여 구성하였으며, 그래픽 인터페이스를 이용한 터치스크린 방식으로 설계함으로써 사용의 편리성을 추구하였다.

### Abstract

I proposed wireless ordering system using Qt embedded. This system will make the good use of real life applications and make fit for practical use. I embodied data input, transfer, gathering and back-transfer on wireless network using TCP/IP protocol on QSocket provided by Qt. Server environment is Linux 9.0 and gcc-2.95.3 on PC, and three target-boards consist PXA255 processor with wireless-LAN card which communicate through wireless access pointer. Software tools on target-board are gcc, tmake, qt-X11, qt-embedded and etc. I designed touch-screen interface for user convenience.

## I. 서론

임베디드 시스템은 대개 특수한 몇 가지의 기능만을 수행하도록 만들어진 컴퓨터 시스템을 말하며, 대개는 큰 하드웨어나 기계장치 내에 포함된다. 임베디드 시스템은 특별히 정해진 일을 처리하므로 최적화하여 설계할 수 있으며 제품의 크기나 가격을 줄일 수 있고, 신뢰성과 성능을 향상시킬 수 있다. 일부 임베디드 시스템은 대량 생산함으로써 경제적인 효과를 얻을 수 있다.

임베디드 시스템에 적용되는 소프트웨어(펌웨어)는 일반적으로 하드웨어적인 제약 때문에 하드디스크에 저장하지 않고 ROM이나 플래시 메모리에 저장한다. 또한 키보드가 없거나 아주 작으며, 모니터가 없는 경우도 있으며, 메모리도 제한되어 있다.

이러한 제약사항으로 인하여 실제의 임베디드 시스템은 서버에서 개발하여 타겟보드로 이식하는 방법을 사용한다. 또한 사용의 편리성을 고려하여 스크린이 필요한 경우 터치스크린을 사용하는 경우가 많으며, 이런 사항들이 모두 고려되어야 한다. 본 논문에서 구축된 시스템도 마찬가지로 터치스크린을

사용하였으며, 최소한의 기능만을 가지는 시스템으로 구성되어 실생활에 이용될 수 있도록 구현하였다.[1]

본 논문에서 사용한 타겟보드는 ARM 계열의 인텔 PXA255를 사용하는 보드이며, 운영체제는 리눅스를 사용하였다. 리눅스의 매력은 소스가 공개되어 있어 사용자가 원하는 기능을 추가 또는 삭제가 자유롭다는 것이다. 제안한 시스템은 데이터 수집, 전송, 처리 등의 일을 하므로 운영체제가 사용되어야 하며, 무료로 사용할 수 있다는 장점을 가진 리눅스를 사용하였다. 이 외에도 임베디드용 운영체제는 Windows CE, Symbian[2], 3Com Palm, pSOS, QNX, VxWorks 등 다양하다. 또한 무선센서 네트워크를 사용하는 임베디드 시스템에서는 TinyOS[3]를 기반으로 하여 nesC가 언어로 사용된다.[4,5,6,7,8] 임베디드 시스템에서는 사용목적에 맞는 적당한 운영체제, 개발언어, 하드웨어 등을 다양하게 선택할 수 있다.

Trolltech사에서 제공하는 GUI 라이브러리인 Qt/Embedded를 사용하여 그래픽 인터페이스를 구현하였다. 이 라이브러리는 X-windows 없이 커널에서 제공하는 프레임 버퍼를 이용하여 그래픽 장치 즉, 터치스크린을 장치에 접근한다.

각각의 타겟보드는 무선 액세스 포인트를 통하여 무선 인터넷으로 연결하며, 수집된 데이터를 전송하여 해당하는 작업을 수행한다.

## II. 개발 시스템

### 1. Qt Embedded

리눅스 PDA의 GUI 표준 경쟁 가시화 : 리눅스 PDA가 표준으로 부상하면서 리눅스 PDA의 GUI 표준 다툼도 치열해질 전망이다. GUI는 운영체제와 애플리케이션을 연결해 주는 다리 역할로, GUI가 다르면 운영체제와 애플리케이션이 서로 호환되지 않아 하드웨어 개발 업체나 애플리케이션 개발 업체, 사용자에게 큰 불편을 가져다준다. 현재 리눅스 PDA 시장은 GTK와 Qt/임베디드 방식이 경쟁을 벌이고 있다.

Qt/Embedded의 강점은 그림 1.에서 보는 바와 같이 C++로 개발돼 유연성이 높으며 다양한 OS 플랫폼을 지원하며, 애플리케이션 개발이 용이하다는 점이다. 즉, C++ 기반의 객체지향 클래스 라이브러리를 사용함으로써 캡슐화 및 상속을 이용한 기능 확장이 가능하며, C++를 확장한 Signal/Slot 메커니즘을 사용하여 객체의 신호를 다른 객체의 슬롯에 연결할 수 있다. 또한 Qt/Embedded는 모바일 단말기를 위해 Trolltech에서 상용화된 GUI로 신뢰성이 검증된 상태다. 그러나 윈도우 CE에 비해 50% 정도 저렴한 가격이지만 라이선스 비용이 필요하다. 다만 개발용으로 배포버전을 사용할 수 있다.

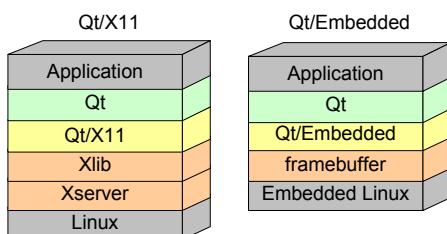
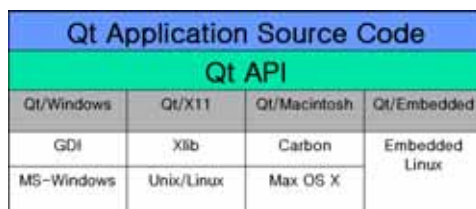
이에 반해 GTK 방식은 무료로 사용할 수 있다는 것이 강점이다. 또한 OOP 프로그래밍이 가능하며, X 윈도우 애플리케이션 개발이 손쉽다. 그러나 데스크톱에서 사용하는 GUI(GNOME)를 채택해 PDA에 탑재하기에는 다소 무거운 감이 있다.

현재 미지리서치와 일본 샤프의 자우루스가 Qt/임베디드 방식을 채용하고 있어 표준 선정에 박차를 가하고 있다. 특히 미지리서치는 이미 삼보컴퓨터, HNT, 게임파크, CI정보기술, 이센스테크놀로지 등 리눅스 PDA 개발업체에 자사의 '리누엣'을 제공하고 있으며, 보라테크와 한컴리눅스에서도 Qt/임베디드 기반의 임베디드 리눅스용 오피스 제품을 내놓고 있다. 특히 한컴의 임베디드 리눅스용 오피스 제품은 이미 샤프의 자우루스에 탑재되어 사용 중이다.

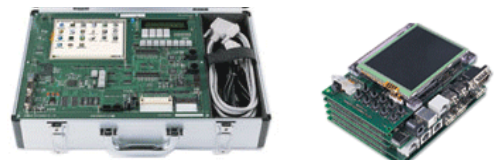
지메이트의 리누피를 중심으로 한 GTK 진영은 GTK를 채용한 요피가 출시됨에 따라 GTK에 대한 재평가가 이뤄지고 있는 실정이다. 이를 입증이라도 하듯, Qt/임베디드 기반의 리누엣을 개발한 미지리눅스도 GTK 방식의 채택을 검토하고 있다고 한다. 또한 ETRI의 내장형 소프트웨어 연구팀도 GTK 방식의 내장형 표준 RTOS를 개발하고 있어 조만간 임베디드 리눅스의 GUI 통일안을 제시할 것으로 보인다.[9]

### 2. 타겟 보드

타겟 보드는 인텔 프로세서인 PXA255 임베디드 프로세서를 장착하고 있는 시중 제품을 사용하였다. PXA255는 첨단기거나 임베디드 프로세서 관련 교육에 국한되지 않고 개발 기술을 실 제품개발에 적용할 수 있는 수준의 프로세서이다. 32MByte의 플래시 메모리, 128MByte의 SDRAM, 1Mbyte의 SRAM을 장착하여 풍부한 메모리환경에서 응용 프로그램을 구현할 수 있다. 6.4인치 컬러 TFT LCD가 장착되어 있어 다양한 형태의 멀티미디어 응용 프로그램에 활용할 수 있으며 터치스크린 환경 또한 함께 제공하므로 PDA 응용 프로그램 개발 환경으로 이용할 수 있습니다. 기타 다양한 외부 인터페이스를 제공한다. 무선인터넷을 이용하기 위하여 무선 LAN카드를 추가하여 구현하였다.[10]



▶▶ 그림 1. Qt/X11과 Qt/Embedded의 구조



▶▶ 그림 2. 타겟 보드

### 3. 전체 시스템 구성

- ① 유무선 공유기를 이용하여 휴대용 소형 무선 단말기를 이용하여 주문데이터를 전송하면 실시간으로 모니터 단말기와 서버로 필요한 정보를 전달한다.
- ② 전달 받은 주문정보를 모니터 단말기에 출력하고, 서버에서는 해당 데이터를 데이터베이스에 추가하고 필요한 계산을 한다.

- ③ 모니터 단말기에서는 실 시간 입력된 내용을 출력하고 관리할 수 있게 한다. 또한 어느 휴대용 단말기로부터 데이터 입력되었는지 보여준다.
- ④ 처리된 결과를 휴대용 단말기로 전송한다.



▶▶ 그림 3. 전체 시스템 구성

### III. 무선 주문 시스템 구현

#### 1. Qt TCP/IP 소켓 프로그래밍

Qt에서 tcp/ip를 사용하기 위해서 QSocket 클래스를 사용한다. QSocket는 버퍼되는 TCP 연결을 제공한다. 이것은 블로킹되지 않는 완벽한 QIODevice를 제공하며 소켓에 한정되는 코드를 가지는 QIODevice의 API를 변경 또는 확장한다. QIODevice로부터 상속받은 함수들, 예를 들면 connectToHost(), bytesAvailable(), canReadLine()와 같은 함수들이 일반적으로 많이 사용된다. connectToHost()는 지정된 호스트 이름을 가진 서버로 연결해 준다.

본 논문에서는 QSocket를 기반으로 제공되는 표 1의 함수들을 사용하였으며, 기타 데이터 전송 및 슬롯 관련된 함수를 표 2, 3, 4와 같이 정의하여 사용하였다. 이들 함수들은 TCP/IP를 사용하며, 클라이언트와 서버 간에 데이터를 실시간으로 연결하여 전송하도록 한다. 클라이언트가 새로 접속할 때 마다 클라이언트 소켓 정보를 저장해 두었다가 브로드캐스팅이 필요한 경우에 접속된 클라이언트 포인터 모두에 전송을 하게 된다.

[표 1] QSocket에서 제공하는 시그널

함수	기능
hostFound()	connectToHost() 함수 호출 후에 상대 호스트가 있다면 발생
connected()	상대 호스트와 연결이 이루어지면 발생
delayedCloseFinished()	연결이 끊어지면 발생
connectionClose()	연결 끊기를 요청했을 때 바로 끊기지 않고 지연을 가지고 끊기를 시도하다가 실제로 끊어 졌을 때 발생
readyRead()	읽어 들일 자료가 있다면 발생
bytesWritten (int nbytes)	nbytes 만큼 전송이 되었음을 알려주는 시그널
error(int)	에러가 발생하면 알려주는 시그널

[표 2] 클라이언트의 슬롯 준비 사용자 정의 함수

시그널 이름	슬롯 함수
connected()	socketConnected()
connectionClosed()	socketConnectionClosed()
readyRead()	socketReadyRead()
error(int)	socketError(int)

실제 전송 패킷은 QString으로 이루어지며, 스트링 가운데 필드를 구분하기 위하여 식별자를 두어 구성하였다. 이 식별자는 응용에 따라 여러 가지로 구성하여 사용할 수 있다.

[표 3] 사용자 정의 슬롯 함수

함수	기능
ServerConnect()	클라이언트가 서버에 접속을 요청 할 때, 호출되는 슬롯 함수
socketReadyRead()	데이터를 전송 받을 때 호출
socketConnected()	서버에 연결되었을 때 메시지 출력
socketConnectionClosed()	서버에 연결이 끊어 졌을 때 메시지 출력
socketClosed()	서버에 연결되지 않았을 때 메시지 출력
socketError(int)	에러 메시지 출력

[표 4] 서버 사용자 정의 클래스 및 함수

클래스/함수	설명
ClientSocket : public QSocket	클라이언트 접속
SimpleServer : public QServerSocket	접속을 알려주는 시그널 발생
MyWidget::newConnect (ClientSocket *s)	클라이언트가 새로 접속 할 때마다 ClientSocket 정보 저장

#### 2. 시스템 구현 및 시험

앞에서 제안한 내용을 바탕으로 실제 무선 주문시스템을 구현하였다. 구현된 시스템은 음식점에서 음식을 휴대용 단말기를 이용하여 주문을 받고 실시간으로 주방으로 전달하며, 서버(카운터)에서 모든 정보를 데이터베이스에 저장하여 계산이 이루어지도록 하였다.

그림 4는 휴대용 단말기의 터치스크린 인터페이스를 보여준다.



▶▶ 그림 4. 터치스크린을 사용한 휴대용 단말기 구현

그림 5와 그림 6은 휴대용 단말기로부터 전송된 데이터를 받아 실시간으로 화면에 디스플레이 해 주며, 데이터베이스 처리 및 계산을 수행한다. 그림 5의 모니터링 단말기는 휴대용 단말기로부터 수신된 데이터 작업이 완료되면 다시 휴대용 단말기로 완료 메시지를 전송하게 된다.

그림 6의 관리 서버는 휴대용 단말기와 모니터링 단말기로부터 수신된 데이터를 데이터베이스로 관리하며 각종 통계 및 계산 작업을 수행한다.



▶▶ 그림 5. 터치스크린을 사용한 모니터링 구현



▶▶ 그림 6. 관리 서버

향후 과제로 남아 있다.

본 논문에서 제안한 시스템은 실제 구현된 내용뿐만 아니라 다양한 분야, 예를 들면 실시간 재고관리 등 무선 네트워크 또는 무선 인터넷을 필요로 하는 분야에 적용할 수 있다.

#### ■ 참고 문헌 ■

- [1] 양영중, 조진희, 하수정, 차정은, 임베디드 시스템 개발방법론 및 재사용 체계, ETRI 전자통신동향분석 제21권 제1호, pp82-93, 2006년 2월.
- [2] Symbian, <http://www.symbian.com/>
- [3] TinyOS, <http://tinyos.re.kr/>
- [4] Ken Arnold, Embedded Controller hardware design, LLH technology publishing, 2001.
- [5] Ashfaw A. Khan, Practical linux programming: device drivers, embedded systems, and the internet, Charles river media, 2002.
- [6] Anthony J. Massa, Embedded software development with eCOS, Prentice-Hall, 2002.
- [7] Michael J. Pont, Embedded C, Addison Wesley, 2002.
- [8] Barnett, Cox and O'Cull, Embedded C programming and the Atmel AVR, Tomson Delmar learning, 2003.
- [9] 한국표준협회-PDA, <http://www.it-standards.or.kr/>
- [10] Jason Hill and David Culler, A wireless embedded sensor architecture for system-level optimization.

## IV. 향후 과제 및 결론

제안된 시스템은 실제로 구현하여 시험해 본 결과 많은 응용 분야에 이용할 수 있다는 결과를 얻었다. 구현해 본 레스토랑 실시간 주문 시스템은 일부에서 사용되고 있는 곳도 있으나 가격 측면에서 경쟁력을 갖기 위해서 리눅스를 기반으로 하는 임베디드 시스템을 사용하였다. 불필요한 요소를 제거함으로써 저렴한 가격에 제공할 수 있을 것으로 판단된다. 구현된 시스템에서는 서버도 리눅스로 구현하였으나 일반사용자에게 익숙하지 않는 점과 호환성 및 기존 자원을 활용하는 측면을 감안하여 윈도우 환경에서도 구현이 가능하도록 하였다.

현재 시스템에서의 문제점은 휴대용 임베디드 및 모니터링 임베디드 시스템의 메모리의 제약으로 필요한 시스템과 관련된 부분을 제거하여 반드시 필요한 부분만 탑재된 상태이다. 즉, 제안 및 구현된 시스템을 최적화된 알고리즘을 구현하여 실제 상용화 하였을 때의 시스템적인 문제를 해결하는 것이