

---

# Embedded Linux를 기반으로 한 Gas 센서용 ARM11 플랫폼 구현에 관한 연구

안종찬\* · 김영길\* · 나상신\*

Implementation of ARM11 Platform for Gas Sensor Based on Embedded Linux

Jong-Chan Ahn\* · Young-Kil Kim\* · Sangsin Na\*

## 요 약

본 논문에서 진행한 연구는 고속의 ARM 아키텍처 기반의 ARM11 코어를 적용한 최신 MPU인 S3C6400을 이용한 플랫폼을 구현하고 구현된 플랫폼에 OS로서 리눅스를 포팅하여 Handheld 타입의 가스 센서용 플랫폼 구현에 관한 연구이다. 무선으로 데이터를 전송받기 위하여 블루투스를 이용한 네트워크 환경을 구축하였다. QT/embedded를 사용하여 리눅스 상에서 유연한 어플리케이션 개발 환경을 구축하여 관련된 프로그램을 작성하였다.

## ABSTRACT

This paper focuses on the implementation of handheld type ARM11 Platform for gas sensor based on Embedded Linux OS. The S3C6400, which is ARM11 architecture based, is the key component of the ARM11 Platform. Bluetooth is adapted to consist the network for the wireless transmission of environmental data between a sensor node and handheld type ARM11 Platform. Linux is ported to the Platform, QT/embedded is used for the application development.

## 키워드

ARM11, 리눅스, QT/embedded, 블루투스, 모바일

## I. 서 론

현대인들은 하루 중 80% 이상을 건물 내에서 생활하고 있어 쾌적하고 건강한 실내공기질의 확보는 매우 중요하다. 에너지절약 차원의 건축물 설계·시공은 환기부족 등으로 실내공기 관리여건은 점점 더 악화되고 있다. 환경오염 및 가정 또는 실내 작업장 내의 환경 상태를 감지할 수 있는 장치에 대한 요구가 증가되고 있는 실정이다. 즉, 주거 및 사무 환경 하에서 일반 건물 안에서 이러한 장치와 솔루션의 필요성이 증대되고 있는 추세이다.[9]

한편으로는 임베디드 플랫폼의 OS 중에서 리눅스의 비중이 커져가고 있는 추세이다. 리눅스는 자유 소프트웨어와 오픈 소스 개발의 대표적인 표본으로써 다중 사용자, 다중 작업 (Multi tasking), 다중 스레드를 지원하는 운영체제이다. 리눅스는 벤더의 독립성과 적은 개발비, 보안성과 안정성이라는 측면에서 여타 다른 임베디드 OS 못지않게 뛰어나다. 리눅스는 개인용 컴퓨터에서부터 슈퍼컴퓨터는 물론 휴대전화, 내비게이션과 같은 임베디드 시스템까지 광범위하게 이용되며 그 수요가 증가하고 있다.

이러한 요구를 충족시키기 위하여 본 논문에서는 임베디드 리눅스를 기반으로 한 휴대용 가스 센서용 플랫폼 구현에 관한 연구를 진행하고 이를 구현하는 데 초점을 두었다. ARM 11 코어를 적용한 S3C6400 어플리케이션 프로세서를 사용하여 휴대용 임베디드 플랫폼을 구성하고 리눅스를 포팅을 한다. 관련된 유저 어플리케이션 프로그램은 QT를 사용하여 작성하며 블루투스를 사용하여 가스 센서와 휴대용 임베디드 플랫폼 사이에 네트워크를 구축한다.

본 논문에서는 ARM11 코어, 리눅스, 블루투스 등에 관한 연구를 언급하고 휴대용 가스센서 플랫폼의 개요와 동작과 그 실험에 대한 결과를 서술할 것이다.

## II. 제안한 플랫폼의 기반기술

### 2.1 The System with ARM11 Platform

그림 1은 본 논문에서 구현한 임베디드 리눅스를 기반으로 한 휴대용 가스 센서용 ARM11 플랫폼 시스템의 개요를 나타내고 있다. 사용자가 휴대용 ARM11 플랫폼

으로 주변 센서의 정보를 블루투스를 통하여 얻고자 하면 플랫폼 즉 리더기는 주변의 환경 정보를 주기적으로 얻는 센서들과 블루투스 연결을 시도하고 네트워크가 구성되면 복수의 센서로부터 데이터를 얻어와 사용자에게 표시하여 준다.

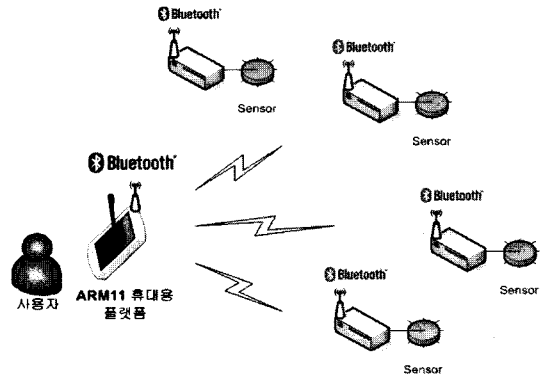


그림 1. 시스템 구성도  
Fig 1. Outline of the Implemented System

### 2.2 ARM11 코어의 주요특징

ARM11 Microarchitecture 는 효율적으로 고성능을 제공하도록 설계되어 코어 클럭의 향상은 물론 전체적인 퍼포먼스의 향상을 가져왔다. 그리고 여러 디바이스에 대한 컨트롤러를 내장하여 시스템의 확장성을 향상시켰다.

ARM11 Microarchitecture 는 종전의 ARM10의 6단계 파이프라인에 비해 8단계 베이스 파이프라인으로 바뀌었다. ARM11 Microarchitecture 의 8단계 파이프라인은 종래의 ARM 코어에 비해 40%의 throughput 향상을 실현시켰다. 또한 8단계 파이프라인에서는 8개의 처리 스테이지가 동시에 실행된다. 밑의 그림2는 이와 같은 파이프라인을 보여 준다. 그러나 단계수가 많은 파이프라인 구조는 시스템에 지나친 지연 혹은 대기시간을 초래하기 때문에 효율성이 떨어질 수 있다. 일반적으로, 단계수가 많은 파이프라인에서는 일부의 명령이 앞의 명령의 결과에 의존하기 때문에 지연될 수 있다. ARM11의 파이프라인은 포워딩을 완전히 활용함으로써, 이런 지연을 막을 수 있다. 또한, 단계수가 많은 파이프라인을 가진 프로세서에서는 파이프라인에서 처리되는 명령의 원활한 흐름에 인터럽트가 일어날 경우, 예를 들면 분기 명령이 발생했을 때에도 성능이 떨어질 수 있다.

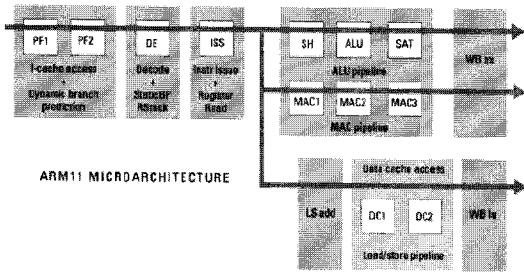


그림 2. ARM11 파이프라인 구조  
Fig 2. ARM11 Pipeline Architecture

ARM11의 파이프라인은 명령의 흐름을 예상하는 분기예측을 사용하여 이 지연을 막는다. 포워딩과 분기예측은 파이프라인의 stall을 줄여 효율성을 유지한다. stall이란 다음 명령어가 아직 파이프라인을 통과중이기 때문에 프로세서가 기다리지 않으면 안 되는 상황을 가리킨다. 이상과 같은 파이프라인의 최적화에 의해, 유효 대기시간은 ARM9TM Microarchitecture의 5단계 파이프라인과 같으면서 훨씬 높은 throughput을 실현한다. [6]

### 2.3 Linux Kernel Structure

커널이란 운영체제의 핵심을 이루는 요소로서 컴퓨터 내의 자원을 사용자 프로그램이 사용할 수 있도록 관리하는 프로그램이다.

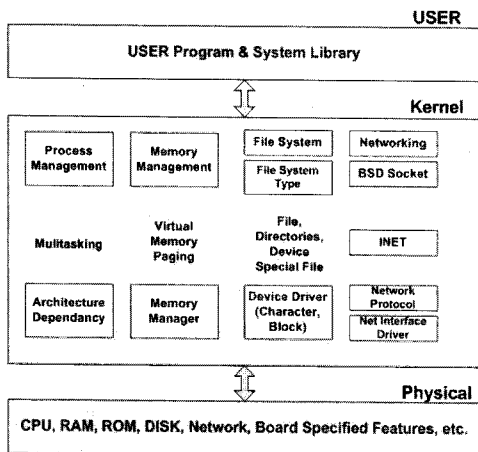


그림 3. Kernel의 기본 구조  
Fig 3. Basic Structure of Kernel

커널에서 제공하는 기능은 크게 보면 프로세스의 관리, 파일 시스템, 메모리 관리, 네트워크가 있으며, 사용자 프로그램은 이러한 기능들을 정해진 규칙에 따서 커널에 요구하게 되며 커널은 이러한 요구를 만족시켜 주어야 한다. 그림 3은 리눅스 커널의 기본구조를 보여준다. 리눅스 커널은 커널의 소스를 수정함으로써 새로운 프로세서로의 포팅이 자유롭다. 리눅스 커널 소스의 트리 구조는 다음의 그림 4와 같다. [3]

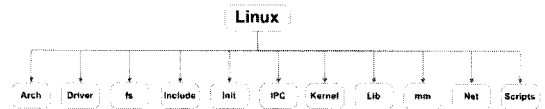


그림 4. 리눅스 커널 소스의 트리구조  
Fig 4. Linux Kernel Source Tree Structure

아래 표 1은 커널 디렉토리 내용을 설명한 것이다. [4]

표 1. 커널 디렉토리에 관한 설명  
Table 1. Description of Kernel Directory

디렉토리	설 명
Arch/	플랫폼에 의존적인 코드
Driver/	하드웨어를 제어하는 디바이스 드라이버
Fs/	파일 시스템 구현
Include/	커널 소스 헤더 파일
Init/	커널 초기화 코드
Ipc/	시스템 V 프로세스 사이의 통신
Kernel/	프로세스, 타이밍, 프로그램 실행, 시그널 등 커널 핵심코드
Lib/	커널 내부적으로 사용하는 범용 Library
Mm/	메모리 관리
Net/	네트워크 프로토콜 스택
Scripts	커널 이미지를 만들기 위해서 사용되는 셸 스크립트 및 프로그램

### 2.4 QT/Embedded

QT의 장점은 강력하고, 사용하기 쉬우며, 플랫폼에 구애받지 않는 GUI 툴킷이라는 점이다. QT는 사용자 인터페이스 클래스는 물론 파일 시스템 액세스, 데이터와 시간값 다루기, 네트워크 프로그래밍에 관한 이식 가능

한 지원을 특징으로 한다. QT는 이미 개발자가 필요로 하는 코드를 가지고 있으므로 QT를 가지고 프로그래밍을 하게 되면, 개발자는 플랫폼 독립 코드에만 국한된 약간의 코딩 작업만 하면 되는 간편함이 있다.

QT는 다음과 같은 특징이 있다. C++로 만들어졌으며 그 구조는 MFC와 비슷하다. 따라서 기존의 C 개발 방식에 비해서 훨씬 용이하고, 기능 확장이 편리하여서 생산성이 높다. 객체 지향성을 강조한 QT 라이브러리는 모듈화와, 구성물을 재활용 가능하도록 초점을 맞춰서 설계되었다. QT는 멀티 플랫폼을 지원하는데 이는 소스의 크로스 컴파일을 통해 어떠한 플랫폼이든 이식이 가능하다는 의미이다. QT에서는 Signal & Slot인 이벤트 처리 모델을 두어 이것을 사용해서 객체 간의 통신이 가능하게 한다. 이로써 복잡한 Callback 함수를 만드는 부담을 줄일 수 있는 것이다. [7]

QT는 어플리케이션 외에도 QT/Embedded와 Qtopia라는 것이 있다. QT/Embedded는 메모리와 CPU 등의 리소스 부담이 많은 임베디드 시스템에 탑재되기 위하여 개발된 것으로 모든 GUI function을 제공하며, 현재 임베디드 리눅스 장비에 사용된다. Qtopia는 QT/Embedded에 기반을 둔 PDA, 팜톱 컴퓨터 등의 기기에 탑재되는 Windows Management로서 리눅스에서 Gnome과 KDE와 같은 기능을 한다. 본 논문에서는 QT를 이용하여 휴대용 가스 센서용 플랫폼에 해당하는 어플리케이션을 구현하고, Qtopia를 이용하여 리눅스 상의 윈도우 매니저를 포팅하였다. [2]

### III. 네트워크 기반기술 블루투스

#### 3.1 블루투스 내역 및 특징

블루투스는 ‘근거리(Short Range), 저전력(Low Power), 저가(Low Cost)로 특화된 무선 인터페이스’라고 할 수 있다. 사용하는 주파수 대역이 2.4GHz의 ISM(Industrial Scientific Medical)밴드이다. ISM 밴드를 사용함으로써 얻을 수 있는 장점은 별도의 허가권을 획득할 필요가 없다는 점이다. 이 밴드의 범위는 24000 ~ 2.4853 GHz이고, 다른 장치와의 간섭을 피하기 위해 상하에 가드 밴드(Guard Band)를 두어 실제 이용한 주파수 대역은 2.402 ~ 2.480 GHz이다. 또 이러한 주파수 대역을 대역폭 1MHz의 79개의 채널로 나누어 각 채널을

호핑하는 방식을 택하였다. 밑의 그림 5는 블루투스 주파수 대역과 채널 배치 및 전송 특성을 요약하여 나타낸 그림이다.

Transmitter Characteristics

Power Class	Maximum Output Power (Pmax)	Nominal Output Power	Minimum Output Power*	Power Control
1	100 mW (20 dBm)	N/A	1 mW (0 dBm)	Pmin<+4 dBm to Pmax Optional: Pmin** to Pmax
2	2.5 mW (4 dBm)	1 mW (0 dBm)	0.25 mW (-6 dBm)	Optional: Pmin** to Pmax
3	1 mW (0 dBm)	N/A	N/A	Optional: Pmin2** to Pmax

Frequency Bands and Channel Arrangement

Lower Guard Band	Upper Guard Band	Regulatory Range	RF Channels
2 MHz	3.5 MHz	400-2.4835 GHz	f+k MHz, k=0...78

그림 5. 블루투스 주파수 대역과 채널 배치 및 전송특성

Fig 5. 블루투스 Frequency Bands and Channel Arrangement and transmitter characteristic

전송속도는 1Mbps ~ 3Mbps이며 출력은 각각 1mW (10m, Class2), 100mW (100m Class1)이다. 주파수 호핑(FHSS: Frequency Hopping Spread Spectrum) 기법을 사용하여 노이즈가 많은 환경에서도 블루투스는 안정된 무선 연결을 보장한다. 네트워크의 구성은 마스터, 슬레이브 형태의 주종관계로 구성되며, 한 대의 블루투스 장치에 동시접속이 최대 7대(ACL 기준)이다. 네트워크의 기본 방식은 TDD(Time Division Duplex)로 625µs의 타임 슬롯을 설정하여 슬레이브마다 할당된 슬롯에서 주파수 호핑을 하며 통신을 한다. [1]

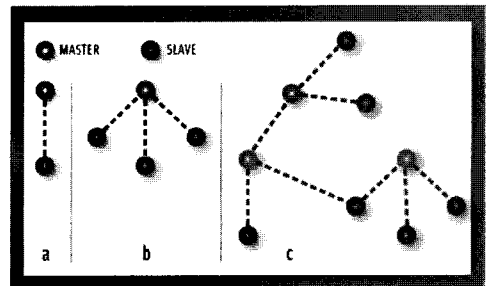


그림 6. 블루투스의 Piconet 구성  
Fig 6. 블루투스 Piconet Connection

마스터와 슬레이브 사이에는 데이터 통신을 위한 ACL(Asynchronous Connection-Less) 링크와 음성 신호

전송을 위한 SCO(Synchronous Connection-Oriented) 링크가 형성된다. 또 Park, Sniff, Hold라는 연결 상태(Connection State)를 설정하여, 8대 이상의 피코넷(Piconet) 구성이 가능하고, 피코넷이 확장된 스캐터넷(Scatternet)이라는 대규모 네트워크 또한 구성할 수 있다. 위 그림 6의 a는 슬레이브가 하나인 경우이며, b는 슬레이브가 여러인 경우이고 c는 스캐터넷을 도식한 것이다. 그림 7은 블루투스의 연결 과정을 그림으로 표현한 것이다.

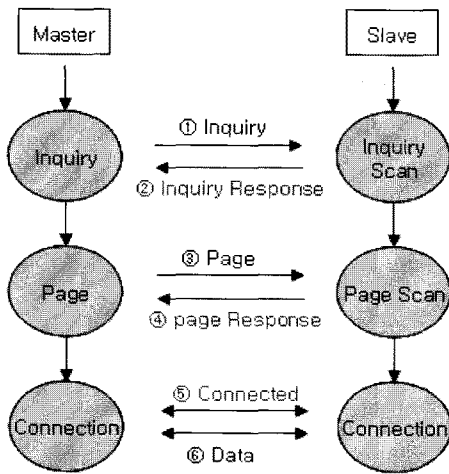


그림 7. 블루투스의 연결 과정  
Fig 7. 블루투스 Connection Operation

#### IV. ARM11 Platform의 구현

구현 시스템은 2장에서 논의한 ARM11 기반의 S3C6400 플랫폼으로 USN 환경에 적합하게 경량의 소형 모바일 리더기 시스템으로 리눅스를 포팅하였다. 또한, 적용 분야인 환경 분야에서 발생할 수 있는 예외 환경들에 대해서도 중앙 관리 서버와의 전송 시스템을 유지하는 대처가 가능하도록 기본적인 무선 인터페이스 블루투스 외에 WLAN 802.11g (SDIO) 및 유선 인터페이스 USB 및 UART에 의한 동작을 지원한다. 위의 그림 8은 구현한 시스템의 내부 블록도이다.

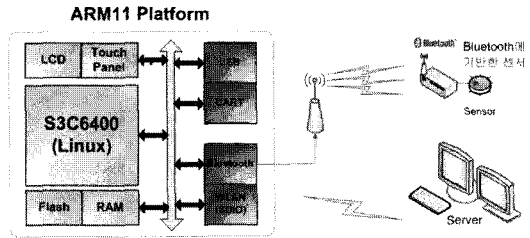


그림 8. 구현 시스템의 블록도  
Fig 8. Block Diagram of the Implemented System

ARM11 코어 기반의 S3C6400으로 400MHz와 533MHz 두 가지 모드로 동작한다. 임베디드 OS로는 리눅스를 포팅하였다. 커널 수정 및 부트로더 작성에서부터 커널 이미지를 보드 상에 올리고 윈도우 매니저로서 QT/embedded 환경을 구축하여 어플리케이션 개발을 하도록 하였다. 시스템의 자세한 세부 사양은 위의 표 2와 같다.

표 2. 시스템 사양  
Table 2. the Specification of the System

구 성	내 용
Main CPU	s3c6400 ARM11 core @ 533MHz OS : Linux 2.6.24 포팅
FlashROM	Flash 64Mbyte
SDRAM	Samsung 64Mbyte
Debugging	Jflashmm, ICE 장비
Data 전송 I/F	USB : Bootloader 다운 UART : 각종 제어 및 디버깅 정보 블루투스 : 센서 네트워크 구성
LCD	Orange 4.7 inch TFT LCD
Sensor Module	ELT H-550 CO2 Sensor (UART, I2C support)
블루투스	FB755AS

#### 4.1 ARM11 core기반의 S3C6400

S3C6400은 삼성이 ARM1176EJF-S CPU 코어를 기반으로 하여 PDA, 2.5G/3G 하이엔드 모바일 폰, 포터블 미디어 플레이어 등에 최적화 되어 있다.

400/533 MHz로 동작하며 64/32bit의 AXI, AHB, APB 로 구성된 internal bus architecture를 가지며 DRAM 포트와 Flash/ROM/ DRAM을 연결 할 수 있는 분리된 2개의 외부 메모리 포트를 가진다. [10]

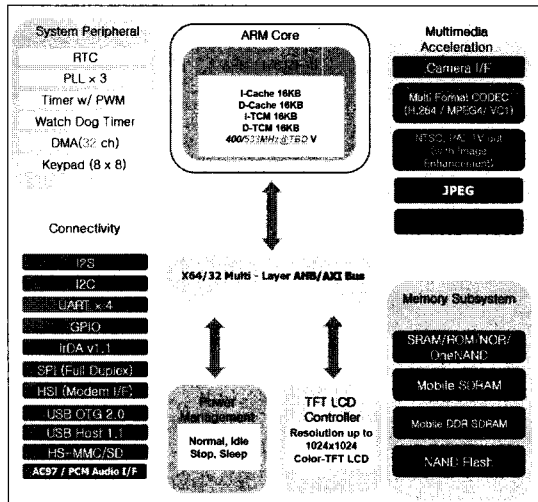


그림 9. S3C6400 블록도  
Fig 9. S3C 6400 Block Diagram

또한 확장성을 용이하게 해주는 많은 하드웨어 peripheral 들을 가지고 있다. 4096X4096의 resolution 을 지원하고 줌, 회전, 컬러 스페이스 컨버전, LCD 컨트롤러 다이렉트 패스 등의 여러 기능을 가진 카메라 인터페이스와 MPEG4, H.264/AVC 등을 지원하는 멀티스탠다드 코덱, JPEG 코덱, 2D 그래픽 Accelerator, 최대 1024X1024 해상도를 지원하는 LCD 컨트롤러, 비디오 포스트 프로세서, TV 인코더, 오디오 인터페이스 등을 갖추고 있다. 또한 외부 확장으로 4채널의 UART, I2C, I2S, 2Ch SPI, HIPI HSI, IrDa 등의 여러 인터페이스와 MMC/SD Host, USB1.1 Host, USB 2.0 OTG 등을 지원한다. 위의 그림 9는 S3C6400의 블록 다이어그램을 보여주고 있다.[5]

### 4.2 Gas Sensor Module

소형의 ELT사의 H-550 CO2 센서 모듈이다. H-550 CO2 센서 모듈은 현재까지 전 세계적으로 가장 초소형, 초경량의 NDIR CO2 모듈로서 홈 네트워크, 환기장치, 컨트롤러 등의 가전제품, 로봇, 자동차에까지 폭넓은 용

용성을 가지고 있다. 또한 모듈탑재가 용이하며 외부기기와 UART, I2C 통신을 지원한다. 자세한 사항은 하기 표 3과 같다.

표 3. 센서 모듈 사양  
Table 3. the Specification of Sensor Module

Item	Specification
Operating Temperature	0°C ~ 50°C
Operating Humidity	0 ~ 95% RH(Non-condensing)
Operating Environment	Residential, Commercial spaces
Sensing Method	NDIR (Non-dispersive Infrared)
Precision	±30ppm ±5%
Measurement Range	0 to 10,000 ppm
Step Response Time (90%)	30 sec
Sampling Interval	3 seconds
Power Input	9 to 18VDC (±10% Regulation)
Current consumption	Normal : 50 mA Peak : 160mA
UART	Asynchronous serial, TTL Level, 38400BPS
I2C	I2C (SCL, SDA)

### 4.3 블루투스 Module

블루투스 모듈인 FB755AS의 모습이다. 블루투스 Specification 2.0을 지원하며 블루투스 Piconets(Point to Multipoint) 구성이 가능하다. 12핀 헤더 타입으로 클래스 1 EDR 중에서 가장 콤팩트한 사이즈로서 본 논문에서 제안하는 휴대용 가스 센서 ARM11 플랫폼 구현에 적합하게 설계가 되어있다. 대기 시에는 평균 8mA 가량의 전류를 소모하며 데이터를 송수신 하는 동작 시에는 최대 40 mA의 소모 전류를 보이고 있어 저전력 설계에 적합한 스펙을 지니고 있다. FB755AS는 UART 인터페이스를 지원한다. 그리하여 센서 모듈 H-550과 UART 통신을 할 수 있어 연결이 용이하다. 자세한 사항은 표 4에 나타내었다.

표 4. 블루투스 모듈 사양  
Table 4. the Specification of 블루투스 Module

Item	Specification
블루투스	블루투스 Specification 2.0
통신거리	100 M
주파수	2.4 GHz ISM Band
감도	-83dBm (Typical)
전송 파워	16dBm(Typical)
크기	27.7 x 20.6 mm
블루투스 profile support	GAP, SPP
Input Power	3.3V
Current Consumption	100 mA (Max)
Operating Temperature	-20°C - 70°C
Comm. Speed	1,200bps - 230,400bps
Antenna	Dipole Antenna
Interface	UART (TTL Level)
Flow Control	RTS, CTS, DTR, DSR support

4.4 구현 시스템의 소프트웨어 구성

본 논문의 플랫폼에 구현된 임베디드 OS인 리눅스에 대한 응용프로그램으로 사용자에게 편리하도록 GUI 환경으로 구성하였다. 소프트웨어의 구성에 있어서 중요한 부분은 시리얼 포트를 통한 블루투스 모듈과의 통신 과정으로서 근처에 있는 센서와 연결된 블루투스 장치를 검색하여 블루투스로 네트워크를 이루어 센서 정보를 시리얼 포트를 통해 받는 것이다.

임베디드 리눅스를 포팅하기 위해서는 하위단의 부트로더가 안정적인 동작을 보장하여야 한다. 또한 플랫폼에 사용하는 아키텍처에 의존적인 부분과 사용하고자 하는 디바이스에 대한 소스 수정을 하여야 한다.

ARM11 플랫폼은 여타 다른 임베디드 디바이스와 마찬가지로 GUI로 구현을 하기 위하여 리눅스에서 비주얼 윈도우를 제공하는 Qtopia라는 윈도우 매니저를 포팅하였다. Qtopia는 리눅스 환경에서 가장 깔끔한 Visual Interface를 제공하지만, 그 용량이 다소 크다. 그렇기 때문에 보다 일반적인 CRAMFS, CROMFS, ROMFS 등에는 포팅을 하기가 쉽지가 않은 것이다. 그

래서 본 플랫폼에는 플래시에 기존 파일 시스템보다 보다 향상된 JFFS2 (Journalling Flash File System v2)에 포팅하였다. [7]

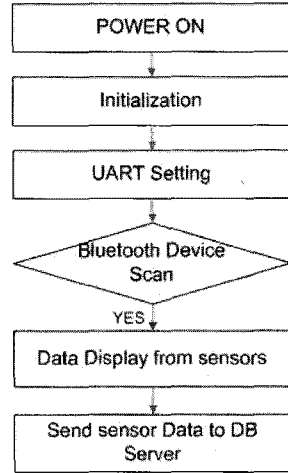


그림 10. 소프트웨어 블록도  
Fig 10. Block diagram of Software

4.5. 실험 결과

그림 11은 그림 1에서 보인 구현된 전체 시스템에서의 개요도에 따라서 실제로 구현한 전체 시스템에 대한 사진으로 본 논문에서 구현한 시스템을 테스트할 실험 환경이다. 아래 부분 좌측에서 구현한 ARM11 플랫폼 및 부착된 블루투스 모듈과 우측의 센서를 볼 수 있고, 그 외에는 서버이다.



그림 11. 구현한 전체 시스템 실험  
Fig 11. The experiment of the system

실험을 위해 직접 제작한 리눅스 어플리케이션으로, 제어 및 디버깅용으로 포트 COM1을 사용하고 있기 때문에 먼저 포트 COM2 및 Baud rate를 설정하고 블루투스 장치 간 자동으로 연결되게 구성을 하였다.

그 후 블루투스 네트워크가 구성이 되면 자동으로 센서에서 넘어오는 데이터가 블루투스 통신에 의해 ARM11 플랫폼의 화면에 디스플레이 되는 것을 확인할 수 있다. 밑의 그림 12와 13은 실험 결과이다.

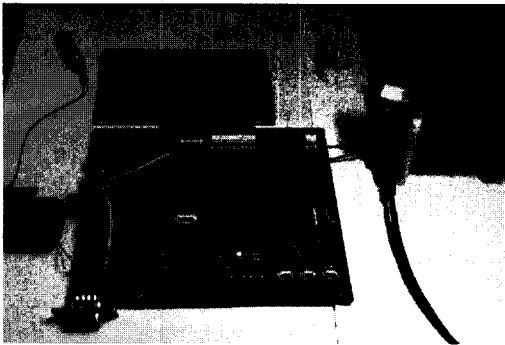


그림 12. 센서 데이터의 인식  
Fig 12. Sensor Data Recognition

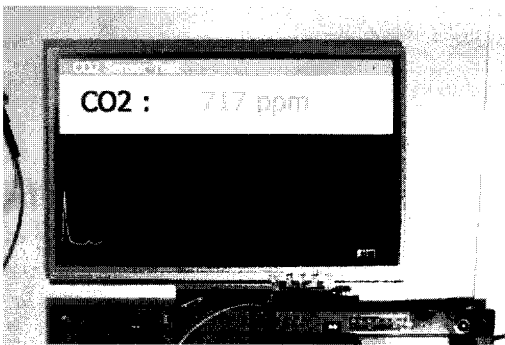


그림 13. 데이터 표시  
Fig 13. Data Display

### V. 결론

본 논문에서는 데이터의 전송을 위한 ARM11 기반의 고성능 S3C6400 플랫폼에 관하여 연구를 진행하고 설계를 하였으며 802.15.1 블루투스의 무선 네트워크를 이용하여 무선 CO2 데이터 전송에 관한 실험을 진

행하였다. 임베디드 리눅스를 기반으로 한 휴대용 가스 센서용 ARM11 플랫폼은 ARM 아키텍처의 최신 버전인 ARM11 코어가 적용된 S3C6400을 이용하여 하드웨어 플랫폼을 고속의 저전력 기반으로 설계하였고, 이러한 ARM11 하드웨어 플랫폼에 리눅스를 포팅하였다. 리눅스 윈도우 매니저인 QT/embedded를 이용하여 어플리케이션 개발환경을 구축하고 관련 어플리케이션을 작성하였다. 플랫폼은 블루투스 뿐만 아니라 WLAN, Zigbee 등의 다양한 통신을 지원할 수 있게 함으로서 이미지, 음성, 동영상 같은 대용량의 정보를 고속으로 전송할 수 있어 유비쿼터스 컴퓨팅 환경에 최적화하도록 설계하였다.

지금까지 본 논문에서 제시한 하드웨어와 통신 실험의 결과를 토대로 더욱 최적화된 프로그래밍 및 하드웨어를 구축한다면, 본 연구에서 구현한 플랫폼보다 더욱 저전력의 고성능 플랫폼이 등장할 것으로 기대하여 본다.

### 참고문헌

- [1] Jennifer Bray 외 1명, "Bluetooth", 홍릉과학출판사, 2001
- [2] 김종덕 "ARM9기반의 Ubiquitous Data Assistant 플랫폼 구현에 관한 연구", 아주대 석사논문, 2006.2
- [3] Christopher Hallinan, "임베디드 리눅스", 정보문화사, 2008
- [4] 타카하시 히로카즈 외 2명 "리눅스 커널 2.6 구조와 원리", 한빛미디어, 2007
- [5] 유호준 "ARM11 기반의 SAW Sensor 리더 플랫폼에 관한 연구", 아주대 석사논문, 2008.2
- [6] ARM.com, "ARM1176JZ(F)-S architecture"  
<http://www.arm.com/products/CPUs/ARM1176.html>
- [7] 이연조, "임베디드 리눅스 프로그래밍", 북토피아, 2007
- [8] linuxfoundation.org by chromatic, "How Linux Supports More Devices Than Any Other OS, Ever", Retrieved on October 31, 2008 from <http://broadcast.oreilly.com/2008/10/how-linux-supports-more-device.html>



- [9] 전재식 (보건환경 연구원), “실내건물의 실내공기 질 관리”, Retrieved on 19 Jan. 2006 [http://www.seoul.go.kr/info/organ/center/healthenv/envsense/air/1228858\\_11630.html](http://www.seoul.go.kr/info/organ/center/healthenv/envsense/air/1228858_11630.html)
- [10] Samsung Application processor s3c6400 [http://www.samsung.com/global/business/semiconductor/productInfo.do?fmly\\_id=229&partnum=S3C6400](http://www.samsung.com/global/business/semiconductor/productInfo.do?fmly_id=229&partnum=S3C6400)
- [11] Bluetooth secure simple pairing user terminology from bluetooth.com White Paper, Retrieved on 9 Aug. 2007 <http://www.bluetooth.com/NR/rdonlyres/1FEB7E6F-575E-42F4-93F7-A13C536B3B5B/0/BluetoothSecureSimplePairingUserTerminologyWhitepaper.pdf>



나상신 (Sangsin Na)

1982년 서울대 전자공학과 (공학사)

1985년 Univ. of Michigan, Ann arbor, 전기및컴퓨터공학과 (공학석사)

1989년 Univ. of Michigan, Ann arbor, 전기및컴퓨터공학과 (공학박사)

1991년 8월 ~ 현재 아주대 전자공학과 정교수

※관심분야 : 정보이론, 자료압축, 디지털 신호처리, 디지털통신

### 저자소개



안종찬(Jong-Chan Ahn)

2007년 아주대 전자공학과 (공학사)

2009년 아주대 전자공학과 (공학석사)

※관심분야: 멀티미디어 통신, 임베디드 시스템, USN, 차세대 무선 네트워크



김영길 (Young-Kil Kim)

1978년 고려대 전자공학과 (공학사)

1980년 한국과학기술원 전자공학과 (공학석사)

1984년 E.N.S.T(프) 전자공학과 (공학박사)

1984년 ~ 현재 아주대 전자공학과 정교수

※관심분야: 멀티미디어 통신, 임베디드, RFID, 차세대 무선 네트워크, 초음파 의료기기