



## A Design and Implementation of The RF Sensing Node Using The ATmega Platform

Worl-Bong Song\*

*Department of Computer Science Engineering, Incheon University*

### ABSTRACT

As the nervous system which create the new information space from connecting the network that make intellectual space, The sensor network technology for core element which realize ubiquitous computing is in the spotlight recently and the research on sensor network which is composed many different sensor node is proceeding actively. As well the RTOS is used for raising the efficiency of system which is concerned about collection and transmission of data from sensor network. In this paper, RF sensing node which smooth communication inter system in the RTOS that used ATmega platform is designed and implemented, and more productively control system is proposed for the control of device from making practice the detection of stability and realizing the light weighted platform. Besides, Typical TinyOS and MicroC/OS are compared and analyzed for the verification. Therefore, The result of this paper is expected to contribute in the development of RF network products which suit about characteristic of ATmega chip in the process of development the RF sensor network products which could display variable information.

© 2015 KKITS All rights reserved

**KEYWORDS :** Ubiquitous computing, Sensor Network Technology, Real Time OS, ATmega chip, light weighted platform, detection of stability

**ARTICLE INFO:** Received 26 February 2015, Revised 10 April 2015, Accepted 10 April 2015.

### 1. 서론

\*Corresponding author is with the Department of Computer Science, Incheon University, 119 Academy-ro Yeonsu-Gu Incheon, 407-772, KOREA  
E-mail addresses: wbsong@incheon.ac.kr,

유비쿼터스 컴퓨팅을 실현하는 핵심요소인 센서 네트워크는 물리적 공간을 지능화하고 이들을 연

결하여 새로운 정보공간을 만드는 신경조직을 구성한다. 이를 근거로 최근 건물 지능화, 도시 주요 시설물 모니터링, 방법 방제 등 여러 분야에서 관심이 높아지고 있으며 더불어 다양한 연구 개발 [1-10]이 진행되고 있다. 또한 센서 네트워크를 통한 데이터의 수집, 전송 등과 관련된 시스템의 효율성을 제고하기 위하여 특성상 RTOS(Real Time OS)[11]가 사용되고 있다. 실시간 운영체제인 RTOS는 처리의 결과와 이 결과가 전달되는 시간에 의해서 좌우된다. 빠르고 예측 가능한 결과를 위해서 실시간 운영체제는 신뢰성(dependability), 예측성(predictability), 동시성(simultaneity), 적시성(timeliness)을 제공함으로써 멀티태스킹과 작업 스케줄링에 집중하여야 하고 빠른 입출력, 최소한의 인터럽트 중지(disabled interrupts) 기간, 최소 메모리 요구 사항의 소형 커널, 개발자가 독자적인 운영체제 기능을 제작할 수 있는 개발 환경 같은 기능을 제공하여야 한다. 이와 관련해서 UC버클리 대학에서 Smart Dust 프로젝트의 일환으로 제작된 임베디드 센서 네트워크 운영체제인 TinyOS[12]는 RTOS(Real Time OS)의 일환으로 necC기반의 센서 네트워크 어플리케이션의 개발을 지원하고, 저 전력 센서 네트워크용 소형 운영체제로 여러 분야에서 활용되고 있으며, 함께 제공되는 necC언어를 통해 event 기반 처리와 효과적인 태스크간 동시성을 지원하고 있다. 또한 기존 컴포넌트들의 손쉬운 연결을 통해 소프트웨어의 유연성과 재사용성이 우수한 소프트웨어 모델을 제시하고 있으며 네트워크 기반의 데이터 전송기술을 기반으로 향후 다양한 네트워크 기반의 제품을 개발할 수 있다는 점에서 관련 연구자들에게도 참조할 만한 사례가 되고 있다. 또한 2009년에 소개되었으며 주로 임베디드 시스템에 적용하기 위해 만들어진MicroC/OS-III( $\mu$ C/OS-III 혹은  $\mu$ C/OS-III 로 통용)[13]는 C 언어 기반으로 작성된 마이크로프로세서를 위한 저비용 우

선순위 기반 선점 형 실시간 운영체제 커널로서 무제한의 태스크, 세마포어, 뮤텍스, 이벤트 플래그, 메시지 큐, 타이머와 메모리 파티셔닝을 제공하며 다중 태스크가 동일한 우선순위 레벨에서 작동하는 것을 허용한다.

본 논문에서는 ATmega 플랫폼[14]을 이용한 소형의 RTOS기반에서 시스템간의 통신에 적합한 무선 센싱 노드를 설계하고 구현하여 안정 및 신뢰성을 검증하고자 하며 각종 상황 변화에 유연하게 대처하고 향후 표준화된 프레임워크 기반 시스템으로 경량화 된 플랫폼을 실현함으로써 장비의 제어에 보다 생산적인 제어시스템 구축을 제안하고자 한다. 따라서 이러한 무선 센싱 노드의 구현은 앞으로 단순정보 획득 및 표출기능을 지양하고 다양한 정보를 표시할 수 있는 무선 네트워크 기반의 제품을 개발하는데 기여할 수 있을 것으로 사료된다.

## 2. 관련연구

### 2.1 ATmega 특징 및 구조

ATmega[14]는 저 전력 8bit RISC 기반 MCU로서 내용별 특징은 다음의 <표 1>과 같다.

ATmega[14]는 RISC 기반 MCU로써, 각 핀은 포트 로 분류되며 포트 A부터 포트 F까지 존재한다. 포트 A, C 는 외부 주소 지정에 사용되며, 포트 B 는 SPI 통신, 포트 D는 TWI, Serial 통신에 주로 사용되고 외부 클럭을 지정할 수 있다. 포트 E는 Serial, 비교기로 사용되며, 포트 F는 ADC 기능을 제공한다. 각 핀은 2개 이상의 기능을 제공하는 경우도 있는데, 이러한 경우는 전기적 신호의 간접 등의 이유로 반드시 1가지 목적으로 사용하여야 한다. 이는 MCU 프로그래밍 과정에서 포트 정의를 이용하여 기능을 정의할 수 있다.

표 1. ATmega의 내용별 특징  
Table 1. The characteristic of ATmega chip

구분	특징
Memory	- EEPROM - 프로그램 적재 공간, 장비 운용 중 데이터 저장 공간, 재사용 가능 - 4K SRAM - 장비 운용 중 사용하는 휘발성 RAM - 최대 64k의 외부 메모리 사용 가능
JTAG interface	- JTAG를 이용하여서 칩의 상세 단계를 디버깅할 수 있음 - EEPROM, SRAM의 직접 데이터 Read/Write 가능
8bit/16bit timer	내부 인터럽트에 사용가능
8bit PWM Channel	PWM 발생 가능, 모터 제어에 유용함
8Channel ADC	-10bit 해상도의 ADC(A/D-Converter)
2개의 Serial, 1개의 SPI, 1개의 TWI	- 외부 장치와 통신하기 위한 다양한 통신 방법 제공 - 각각은 프로그래밍 가능함.
MCU 특화 기능	- Power-On Reset 기능으로 전원 구동 시 바로 작동 가능 - Brown-Out Detection 기능으로 전원이 낮거나 불안정시 시스템 동작을 리셋 상태로 전환 - External/Internal Interrupt - 다양한 Interrupt 제공 가능

<그림 1>에서 포트 A-F는 상태, 방향 레지스터로 제어하고 ADC 입력은 AVCC, AGND, AREF를 이용하여 AREF의 기준전압을 기반으로 값을 측정하며 Program Counter, Stack Pointer를 이용한 메모리 시스템 상태를 제어한다. 또한 8bit Addr./Data 라인을 가지며 인터럽트 카운터를 이용한 내/외부 인터럽트 제어 및 JTAG 기능 지원을 위한 ON-CHIP Debug를 지원하며 주파수 분배로 적절한 클럭을 제공한다.

## 2.2 RTOS

RTOS(Real Time OS)[11]는 정해진 시간 제약을 해결할 수 있는 시스템을 말한다. 최근 운영체제는 컴퓨터의 성능을 향상시키는데 많은 도움을 주고 있다. 그러나 실시간 시스템에서의 성능은 단지 성능 평균 시간에 의해서만 측정 되지는 않는다. 실시간 시스템의 제약조건은 시간 제약 조건 외에도 자원 우선순위, 또는 선행 관계, 태스크 간의 통신 및 동기화 제약 조건이 있을 수 있다.

다음 <그림 1>은 ATmega 칩의 구조를 나타낸다.

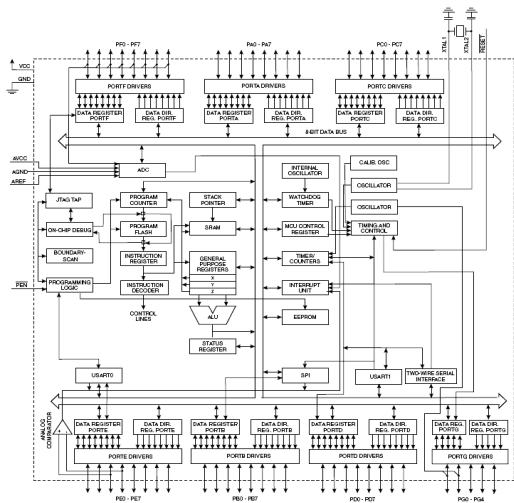


그림 1. ATmega chip의 구조  
Figure 1. The structure of ATmega chip

표 2. 실시간 운영체제의 특징  
Table 2. The characteristic of Real Time OS

구분	내용
1	다중 프로세스 선점 가능(preemptive)
2	예측 가능한 스레드 동기화
3	우선순위에 근거한 선점 형(preemptive) 작업 스케줄링
4	외부 이벤트에 예측 가능한 반응(Hard real-time systems에서는 1 마이크로세컨드, soft real-time system에서는 10밀리세컨드 정도의 반응 시간이 요구됨)
5	빠른 입출력
6	최소한의 인터럽트 중지(disabled interrupts) 기간
7	최소 메모리 요구 사항의 소형 커널

따라서 실시간 운영체제의 가치는 처리의 결과와 이 결과가 전달되는 시간에 의해서 좌우된다. 빠르고 예측 가능한 결과를 위해서 실시간 운영체제는 신뢰성(dependability), 예측성(predictability), 동시성(simultaneity), 적시성(timeliness)을 제공함으로써 멀티태스킹과 작업 스케줄링에 집중하여야 하며 구체적으로는 <표 2>와 같은 기능을 제공하여야 한다. 이의 예로는 TinyOS[12], MicroC/OS[13]등이 있다.

### 2.2.1 TinyOS

TinyOS[12]는 UC 버클리에서 개발된 임베디드 센서 네트워크 OS로서 네트워크에서 각종 센서 데이터를 수집하고, 저 전력, 적은 코드 사이즈로 구동이 가능한 OS이다. 개발 환경 구축은 Cross-Platform간에 컴파일을 하여 목적 파일을 생성하는 환경에 대한 정의로 잘 구축된 개발 환경은 개발 시간을 크게 단축할 수 있으며 개발자로 하여금 정확한 목적 파일을 생성할 수 있도록 한다. TinyOS[12]는 nesC를 사용하여 컴파일 하고 각 플랫폼에 맞는 Makefile이 필요하다.

TinyOS 커널은 한 번에 하나의 어플리케이션만 실행하는 구조로 되어 있으며 전원 인가 후 단순 FIFO 형태의 스케줄러로 작동하고 스케줄러는 전원 공급 시 무한 작동하는 구조로 되어 있다. 따라서 TinyOS 커널 정확한 동작과정과 전원 인가 후 작동되는 구조를 알아야 하드웨어를 제어할 수 있다. TinyOS 커널은 MainM.nc에서 시작하며, 전원 인가 후에 하드웨어, 센서에 대한 초기화, 스케줄러 초기화, Component

들의 초기화, 하드웨어 인터럽트동작, 이벤트 대기 순으로 진행된다. 다음 <그림 2>는 TinyOS의 부팅 후 동작 순서를 보여 주며 <그림 3>은 TinyOS kernel programming의 예를 보여준다.

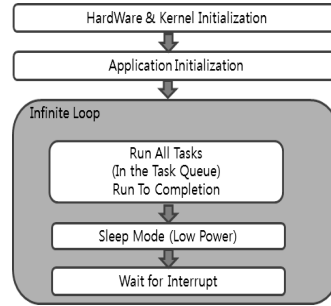


그림 2. 부팅 후 플랫폼 동작 순서  
Figure 2. Operating sequence after booting

```

module MainM
{
    usescommandresult_thardwareInit();
    usesinterfaceStdControl;
}
implementation
{
    int main() __attribute__((C,
    spontaneous))
    {
        call hardwareInit();
        TOSH_sched_init();
        call StdControl.init();
        call StdControl.start();
        _nesc_enable_interrupt();
        for(;;) { TOSH_run_task(); }
    }
}
  
```

그림 3. TinyOS 커널 프로그램  
Figure 3. TinyOS kernel program

### 2.2.2 MicroC/OS

MicroC/OS[13]는 C 언어 기반으로 작성된 마이크로프로세서를 위한 저비용 우선순위 기반 선점형 실시간 운영체제 커널이다. 2009년에 소개되었으며 주로 임베디드 시스템에 적용하기 위해 만들어졌다. 무제한의 태스크, 세마포어, 뮤텍스, 이벤트 플래그, 메시지 큐, 타이머와 메모리 파티셔닝을 제공한다. 사용자는 모든 커널 오브젝트를 실행 시간에 할당한다. 주요 특징은 다음과 같다.

(1) 확장성

μC/OS는 성능 및 확장성을 위해 μC/TCP-IP,

μC/GUI, μC/File System, μC/USB, μC/CAN, μC/Modbus, μC/Bluetooth등의 소프트웨어 패키지와의 결합을 허용한다.

(2) Task monitoring

태스크를 모니터링 하기 위해 스택 증가를 허용하는 기능을 제공한다. 태스크의 크기는 제한되지 않기 때문에 CPU 사용량을 바탕으로 최소의 사이즈를 가져야 한다.

(3) Multitasking

다중 태스크가 동일한 우선순위 레벨에서 작동하는 것을 허용한다. 동일한 우선순위 태스크들이 실행 준비 상태 일 때, μC/OS 는 사용자가 지정한 시간에 맞춰 각각 실행한다. 각각의 태스크는 양자화 된 시간을 정의할 수 있고, 태스크가 그 시간을 모두 채울 필요가 없을 땐 주어진 타임 슬라이스를 내어주게 된다.

(4) Stack

태스크 스택들은 각각의 태스크마다 크기가 자유로운 스택을 제공하기에 더 나은 공간 관리를 허용한다.

(5) Kernel

커널은 255단계의 인터럽트를 관리하며 세마포어 없이 태스크를 시그널링 하는 것이 가능하다.

### 3. RF 센싱 노드 설계

#### 3.1 시스템 구성

습도, 온도, 조도 등의 정보를 수집하고 데이터 처리를 하기 위하여서는 각 센서에게 알맞은 인터페이스를 제공할 수 있어야 하며 센서에서 획득한 값을 해석할 수 있는 컨트롤러가 필요하게 된다. 따라서 MCU를 부착하여 센서정보를 해석하고 이를 적절한 값으로 변경할 수 있는 기능의 보드를 설계 및 구현하였다. 설계에 요구되는 사항은 통

신, 수식 등의 계산을 위하여 속도는 8MHz 이상이 되어야하며, 신호변환을 위하여 ADC(Analog Digital Converter), 데이터전송을 위하여 범용 비동기화 송수신장치(UART) 등의 주변장치가 필요하고 주변장치의 추가 확장을 위한 포트제공 등이 요구된다.

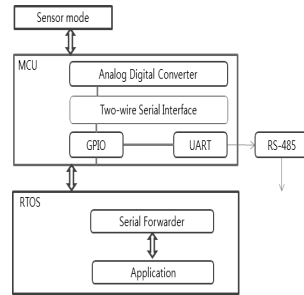


그림 4. 시스템 구성  
Figure 4. System architecture

#### 3.2 BASE STATION

속도, 주변기기, 작동의 용이성, 구입의 용이성, 그리고, RTOS[11] 지원 여부를 기준으로 MCU(통합 제어보드)는 Atmel사의 ATmega128A[14]를 사용하였다.

표 3. ATmega128A 플랫폼기능  
Table 3. The function of ATmega128A platform

항 목	내 용
MCU	ATmega128A(8bit RISC, 14.7456Mhz)
Power	5V
USB Connector	RTOS 이미지 업 로딩
데이터 전송률	250kbps
통신주파수	2.4GHz
안테나	세라믹안테나(실내30M통신가능)
ISP	6Pin ISP(SPI 동작)
Interface	- 8개의 ADC - 2개의 UART (RS-485 변환) - 1개의 TWI - 8개의 GPIO
Debug	UART 1채널

ATmega128A는 16Mhz로 동작이 가능하며, ADC, UART, SPI, TWI 등 주변 통신장치가 있고 128KB의 ROM, 4K의 RAM이 있어 소형 임베디드 환경에 적용하기 알맞다. ATmega128A에 대한 사양은 <표 3>과 같다.

### 3.3 RF Sensor Node

센서 네트워크에서의 무선통신을 담당하는 무선 통신부는 안테나, RF와 modem을 포함하는 물리계층 및 MAC(Media Access Control)프로토콜[4,5]로 이루어진다. 물리계층은 현재 IEEE 802.15.4 프로토콜의 MAC 프로토콜 표준에 상응하는 TI의 CC2420이 센서 네트워크 플랫폼의 RF transceiver로 많이 사용되고 있다. CC2420은 2.45GHz 주파수 대역을 사용하고 데이터 전송률은 250kbps이며 송신 신호의 전력소모를 줄이기 위하여 O-QPSK 변조 후 DSSS 확산 방식을 사용한다.

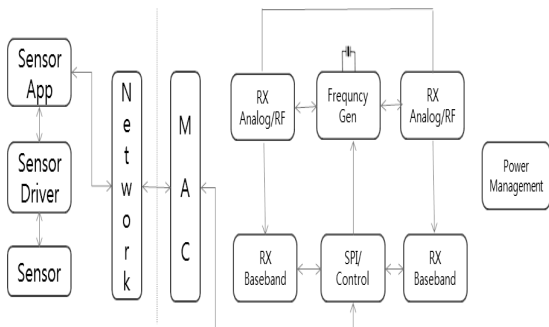


그림 5. 센서 노드의 구성요소  
Figure 5. Structure elements of sensor node

동기 방식으로 동작하는 무선 센서 네트워크 MAC 프로토콜은 동기를 맞추기 위해 주기적으로 Control Packet을 주고받기 때문에 프로토콜 오버헤드가 크며 송수신에 참여하지 않는 노드들도 일정한 시간동안 Listen 상태를 유지하기 때문에 Idle

Listening으로 인한 에너지 소모가 크다. TinyOS에서 채택하고 있는 B-MAC 프로토콜은 비동기 방식으로 동작하는 대표적인 무선센서 네트워크 MAC 프로토콜로써 Low Power Listening(LPL) 기법을 사용한다.

## 4. 시스템 구현 및 실험

### 4.1 시스템 구현

<그림 6>은 개발 시스템의 구성 도를 나타낸다. 요구사항과 같이 ADC 기능이 있어 센서에서 전송하는 데이터를 해석하고 UART를 사용하여 외부로 데이터를 출력하며, RS-485로 변환되어서 데이터를 전송한다. 또한, MCU가 전체 시스템을 제어하는 형태로 되어있다. 전원은 외부에서 안정된 전원을 인가받아서 전체 시스템을 동작시킨다.

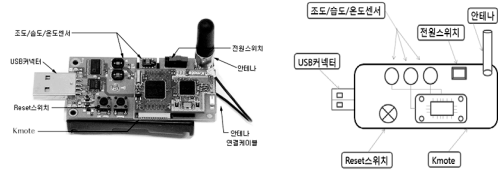


그림 6. 통합 플랫폼 보드  
Figure 6. Complex platform board

무선 센싱 노드의 경우 대표적으로 다음 3개를 선정 하였으며 각 센서별 사양은 다음 <표 4>와 같다.

표 4. 센서별 사양  
Table 4. The specification of sensors

종류	구분	내용
습도	Relative Humidity(RH)	SHT 11
	Range	0 to 100% RH
	Accuracy	±3.5% RH
	Response Time	≤4 sec
	Reproducibility	±0.1% RH
	Resolution	0.03% RH
	Operating temperature	-40°C to 120°C

온도	Temperature(T)	SHT 11
	Range	-40°C to 120°C
	Accuracy	±0.5°C
	Response time	≤20 sec
	Reproducibility	±0.1°C
	Resolution	0.01°C
조도	Photodiode	S1087
	Spectral response range	320~730
	Peak Sensitivity wave ln.	570

### 4.2 실험 결과 및 고찰

본 연구에서는 습도, 온도, 조도 등의 정보를 수집하고 데이터를 처리하기 위하여 MCU를 부착하여 센서정보를 해석하고 이를 적절한 값으로 변경할 수 있는 기능의 보드를 설계 및 구현 하였다. 센서 운영 체제의 비교 분석을 위해 RTOS로는 디바이스에 대한 지원이 가능한 TinyOS 및 MicroC/OS를 사용하였으며, 평가를 위하여 Thread 기반의 신호 변환(ADC) 과정 수행 및 RTOS별 전송 메시지의 수행 시간을 비교 실험하였다. 테스트 방식은 세 개의 센서를 이용하여 OS에 독립적인 보드에 내장된 Timer를 이용하여 비교 측정 하였다. 다음 <그림 7>은 센서 정보의 흐름을 나타내며 <표 5>는 실시간 OS별 전송 수행 시간을 나타낸다.

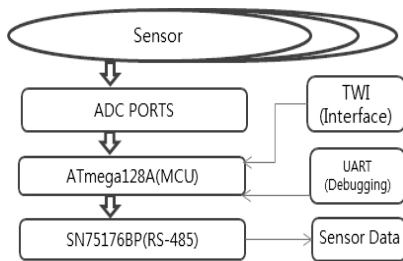


그림 7. 센서 및 통신제어를 위한 블록도  
Figure 7. Block diagram for sensor and communication control

표 5. 실시간 운영체제별 데이터 전송 소요 시간  
Table 5. Data transmission time per RTOS

단위 : ms

구분 센서	Tiny OS					MicroC/OS				
	1	2	3	4	평균	1	2	3	4	평균
온도	115	114	116	115	115.75	92	91	92	93	92
습도	120	122	123	122	121.75	101	101	103	103	102
조도	114	113	114	115	114	91	90	90	91	90.5

<표 5>의 실험 결과에서 볼 수 있듯이 TinyOS는 평균 소요 시간이 115.7ms, 121.7ms, 114ms가 나왔고 MicroC/OS는 평균 92ms, 102ms, 90.5ms가 측정되었다. 또한 각각 수행 빈도에서는 0.01% 오차 범위 내에서 안정적인 전송 소요 시간을 갖는 것으로 검증 되었으며 MicroC/OS가 비교적 낮은 전송 소요 시간을 나타내는 것으로 확인 되었다.

### 5. 결 론

물리적 공간을 지능화하고 이들을 연결하여 새로운 정보공간을 만드는 신경조직인 센서 네트워크는 주요 시설물 모니터링, 방범 방제 등 모든 산업 분야에서 다양한 목적으로 사용되고 있으며 이들 분야에 높아진 관심만큼 다양한 연구 개발이 진행되고 있다. 또한 센서는 원시 데이터 수집만을 수행하여 그 자체를 시스템으로 사용할 수 없기 때문에 센서를 통한 수집 자료의 전송 및 가공하는 과정에서 네트워크를 통한 데이터의 수집, 전송 등과 관련된 시스템의 효율성을 제고하기 위하여 특성상 RTOS (Real Time OS)가 사용되고 있다. 본 논문에서는 ATmega 플랫폼을 이용한 소형의 RTOS기반에서 시스템간의 통신에 적합한 무선 센싱 노드를 설계하고 구현하여 안정성을 검증하고 경량화 된 플랫폼을 실현함으로써 장비의 제어에

보다 생산적인 제어시스템을 제안하였다. 따라서 본 논문의 결과는 다양한 정보를 표시할 수 있는 무선 센서 네트워크 기반의 제품을 개발하는 과정에서 ATmega 칩의 특징에 알맞은 주변장치 사용 및 RTOS의 개발 환경에 대한 이해와 응용을 통한 실시간 운영체제 기반의 다양한 무선 네트워크 제품 개발에 기여할 것으로 사료된다.

## References

[1] Wong Kiing-Ing, *A light-weighted, low-cost and wireless ECG monitor design based on tinyOS operating system*, 6<sup>th</sup> International Special Topic Conference on ITAR, 2007.

[2] Masatuki Nakamura, Atsushi Sakurai, and Shizuo Furubo, Hiroshi Ban, *Collaborative processing in sensor/actuator networks for environment control*.

[3] Jaspal S. Sandhu, Alice MAgogino, and Adrial K Agogino, *Wireless sensor networks for commercial lighting control*.

[4] P. Huang, L. Xiao, S. Soltani, M. Mutka, and N. Xi, *The evolution of MAC protocols in wireless sensor networks*, Communication Surveys & Tutorials, IEEE Vol. 15, Issue 1, pp. 101~120, 2012.

[5] S. C. Kim, and J. H. Jeon, *EEP-B-MAC: Energy efficient & priority-based MAC protocol for energy harvesting wireless sensor networks*, in journal of Korea Institute of Information and communication Engineering, Vol. 17, No 3, 2013.

[6] T. W. Jang, E. J. Kim, and B. S. Lee, *Design of intelligent water vision system*, IWUCA 2010 Conference, 2010.

[7] Watson, M., Stockhammer, and T., Luby, *RTP payload format for raptor forward*

*error correction (FEC)*, RFC6682, August 2012.

[8] Embedded Linux/Microcontroller Project, <http://www.uclinux.org>, 2007.

[9] Stallings W., *Data and computer communications*, PRENTICE HALL, 2011.

[10] GVS, <http://www.e-gvs.co.kr>

[11] FreeRTOS, <http://www.freertos.org>

[12] TinyOS, <http://www.tinyos.net/>

[13] uC/OS, <http://www.micrium.com/>

[14] ATMEL, <http://www.atmel.com>

---

## ATmega 플랫폼을 이용한 RF 센싱 노드의 설계 및 구현

### 송월봉

인천대학교 컴퓨터공학부

---

### 요 약

공간을 지능화하고 이들을 연결하여 새로운 정보공간을 만드는 신경조직으로 유비쿼터스 컴퓨팅을 실현하는 핵심요소인 센서 네트워크 기술이 각광을 받으면서 다양한 종류의 센서 노드로 구성된 센서 네트워크에 대한 연구가 활발히 진행되고 있다. 또한 센서 네트워크를 통한 데이터의 수집, 전송 등과 관련된 시스템의 효율성을 제고하기 위하여 특성상 RTOS(Real Time OS)가 사용되고 있다. 본 논문에서는 ATmega 플랫폼을 이용한 소형의 RTOS 기반에서 시스템간의 통신에 적합한 무선 센싱 노드를 설계하고 구현하여 안정성을 검증하고 경량화 된 플랫폼을 실현함으로써 장비의 제어에 보다 생산적인 제어시스템을 제시하고자 한다. 또한 검증을 위해서는 RTOS의 대표적인 TinyOS와 MicroC/OS를 비교 분석한다. 따라서 본 논문의 결과는 다양한 정보를 표시할 수 있는 무선 센서 네트워크 기반의 제품을 개발하는 과정에서 ATmega 칩의 특징에 알맞은 제품의 개발에 기여할 것으로 기대된다.

---



## 감사의 글

본 논문은 인천대학교의 2014학년도 학술연구조성비를 지원 받음.



**Worl-Bong Song** received the bachelor's degree in electronic engineering from the Soong Sil University in 1975. He received the M.S. degree in the computer engineering from Han Yang University in 1982 and ph.D. degree in the Department of Computer Engineering from Soon Chun Hyang University in 1998. He has been a professor in the Department of Computer Science at Incheon University since 2010. His current research interests include parallel processing, algorithm and software engineering. He is a life member of the KKITS.

*E-mail address* : [wbsong@incheon.ac.kr](mailto:wbsong@incheon.ac.kr)