

# ARM 마이크로컨트롤러 기반 RTD-1000A의 구현<sup>☆</sup>

## The Implementation of RTD-1000A based on ARM Microcontroller

김민호\*                      홍인식\*\*  
Min-Ho Kim                  In-Sik Hong

### 요약

유비쿼터스 응용 시스템에 대한 관심의 증대와 함께 소형화된 임베디드 컴퓨팅 시스템의 필요성은 커지고 있다. 이러한 가운데 ARM 임베디드 프로세서는 기능의 우수성과 높은 활용도로 인해 임베디드 시스템 시장에서 높은 점유율을 보여주고 있다. 본 논문에서는 ARM 마이크로컨트롤러를 이용해 RTD-1000 컨트롤러 구성과 개발을 위한 최적의 방법을 제안하였다. 기존 RTD-1000은 케이블의 단선, 단락, 파손 등의 진단이 가능한 TDR를 탑재하여 구리선을 삽입한 감지관의 누수 및 누유, 파괴 등을 원격으로 감지할 수 있는 기기이다. 실제로 시공되어 현장에서 운영되고 있는 RTD-1000은 시스템 운영에 필요한 범위에 비해 리소스 낭비가 크고 그에 따라 구축비용이 높다는 단점을 가지고 있다. 또한, 발열이 심해 별도의 냉각장치가 요구되며, 하드 디스크와 같은 보조저장장치의 사용으로 고장 발생률과 전류의 소비가 커지는 등의 문제점을 야기하였다. 본 논문에서는 도출된 문제점의 해결 방법으로 ARM 마이크로컨트롤러 기반의 RTD-1000A 임베디드 시스템을 제안하고 시뮬레이션 하였다.

### Abstract

With increase of concern about the Ubiquitous application, the necessity of the computer system which is miniaturized is becoming larger. The ARM processor is showing a high share from embedded system market. In this paper, Ideal method for RTD-1000 controller construction and development is described using ARM microcontroller. Existing RTD-1000 measures distance of disconnection or defect of sensing casket by measuring receiving reflected wave which was sent via copper wire inside the leaking sensing rod. Using this RTD-1000, leakage and breakage of water and oil pipe can be sensed and it reports damage results to the networks. But, existing RTD-1000 wastes hardware resources much and costs a great deal to installation. Also, it needs a cooling device because the heating problem, and has some problem of the secondary memory unit such as the hard disk. So, long term maintenance has some problems in the outside install place. In this paper, for the resolving the problem of RTD-1000, RTD-1000A embedded system based on ARM is proposed and simulated.

☞ keyword : ARM, RTD-1000A, SoC, 컨트롤러(Controller)

## 1. 서론

기존에 획일적이고 한정된 범위에서 컴퓨터를 사용해 오던 것과는 달리, 최근 10년 동안 인터넷의 보급과 함께 기술의 빠른 발전으로 컴퓨터는 사

회 전반에 걸쳐 다양한 요구를 반영 할 수 있는 만능기기로 발전해 오고 있다. 이제 “컴퓨터”는 개인용 데스크톱이나 서버의 의미를 넘어 컴퓨팅 가능한 모든 기기들을 칭하게 되었으며, 사용자는 높은 성능, 정확성, 안정성은 물론이고 좀 더 작고, 편리한 컴퓨터를 원하게 되었다. 이를 위해 시스템을 하나의 칩에 담은 SoC(System on Chip)가 하나의 방법으로 각광 받고 있고, 현재 임베디드 시스템 시장에서 약 90% 이상의 독보적인 점유율을 차지하고 있는 ARM 프로세서는 SoC의 전형을 보여주고 있다. 고성능 저전력의 특징을 가진 ARM 프로세서는 핵심 코어를 중심으로 다양한 방향으로 설

\* 준 회원 : 순천향대학교 대학원 컴퓨터학과(석사과정)  
mindhawk@sch.ac.kr

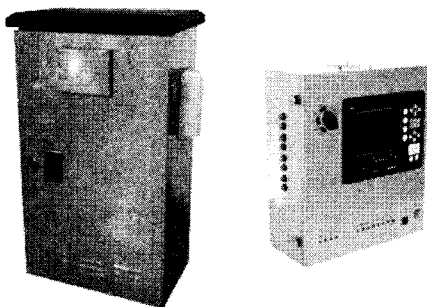
\*\* 정 회원 : 순천향대학교 공과대학 컴퓨터학부 교수  
ishong@sch.ac.kr(교신저자)

[2008/05/13 투고 - 2008/05/21 심사 - 2008/07/22 심사완료]

☆ 본 연구는 21세기 프론티어연구개발사업인 수자원의 지속적 확보기술개발사업단의 연구비지원(과제번호: 4-2-3)에 의해 수행되었음.

계가 가능하므로 다양한 분야에 적용하기 용이한 장점을 가졌다.

본 논문에서 개량하고자 하는 RTD-1000은 선행 연구를 통해 누수탐지 모니터링 시스템을 연구, 개발하였고 현재 현장에서 시공되어 운영되고 있다. 누수탐지 모니터링 시스템은 누수감지관, 모니터링 프로그램, RTD-1000으로 구성되어 있다. 그 중 핵심적 역할을 하는 RTD-1000은 누수감지관에 삽입된 구리도선에 펄스파를 보내고 반사파를 받아서 감지관의 파손으로 발생한 단선 및 고장 점의 거리를 측정하고 측정된 데이터를 서버의 모니터링 프로그램으로 보내는 원격제어감시장치이다. RTD-1000은 TDR(Time Domain Reflectometer), CDMA Modem 그리고 데이터처리와 원격제어를 위한 임베디드 컨트롤러로 구성되어있다[1][2]. 그 밖에 전원장치, 전류 안정기 등을 탑재하고 있고, 누수감지관이 매설된 곳의 지상에 일정한 간격으로 설치되어 있다. 이러한 환경에서 항상 구동되는 기존 RTD-1000의 임베디드 컨트롤러는 범용 PC 사양으로 구성되어 있어 몇 가지의 문제점을 가진다. 첫 번째로 GUI기반 범용 운영체제와 하드디스크와 같은 보조기억장치 사용으로 리소스 낭비가 크고 고장 발생률이 높다. 두 번째로 장치의 외양이 커지고, 전력 소비가 크다. 세 번째로 발열량이 크므로 별도의 냉각장치를 필요로 한다. 결과적으로 임베디드 프로세서 기반의 컨트롤러에 비해 구축비용이 높아지며, 한여름과 겨울에는 그 내구성에 있어 커다란 문제점을 안고 있다.



(그림 1) RTD-1000의 외함과 내함

이러한 문제점의 해결 방안으로 기존 시스템을 분석하여 ARM 마이크로컨트롤러(이하 ARM)를 적용하기 위한 방안을 제안하고 RTD-1000 컨트롤러를 개량한다. 이를 위해서 ARM 기반 전용 컨트롤러 보드를 구성하고 그 위에 전용 프로그램을 구동하기 위한 임베디드 소프트웨어 모듈을 구현하여 RTD-1000A라 명명한다. 이와 같이 본 논문에서는 RTD-1000A를 구현하고 시뮬레이션 한다.

## 2. 관련 연구

### 2.1 RTD-1000(Remote TDR Device-1000)

RTD-1000은 수 나노미터에서 수십 마이크로미터 파장 범위의 전자파를 일정한 간격으로 감지도선에 전송해 반경 수km까지 감지도선이 삽입된 감지관 네트워크를 모니터링 할 수 있는 기기이다[3].

그림 1에서 왼쪽에 위치한 외함은 가로 540mm, 세로 330mm, 높이 920mm 크기로 내함, 전원장치, 전류안정기를 포함하고 있다. 내함에는 TDR 장치와 컨트롤러가 구성되어있다[4]. 외함은 도로 위 교통제어기의 크기와 모양이 흡사하고 누수감지관이 매설된 곳의 지상에 놓여 설치된다. 그러므로 훼손이나 파손의 우려가 높고 보안상 문제점이 있다. 하지만 기존 RTD-1000은 크기의 제약으로 현재 지상에 설치하는 방법이 가장 최선이라고 판단되어 도로 변 적당 위치에 설치되고 있다.

### 2.2 32비트 기반의 ARM 프로세서

#### 2.2.1 특징

ARM 프로세서는 ARM사에서 설계한 코어를 가지고 제작하는 모든 프로세서 및 마이크로컨트롤러를 말한다. 32비트 RISC(Reduced Instruction Set Computer)로 한정된 하드웨어 리소스에도 불구하고 성능이 우수하며, 전력 소모가 작고 가격 경쟁력이 우수한 장점을 가진다. 현재 대부분 시스템에서의 핵심은 프로세서 자체의 속도가 아니라 시스템 전

체의 효율적인 성능과 낮은 전력 소모에 근간을 두고 있는 상황에서 ARM 프로세서는 이에 가장 부합된다[5]. 하지만 ARM 시스템은 비용이나 물리적인 크기의 제약으로 제한된 메모리를 가지기 때문에 코드의 집적도가 상당히 중요한 요구사항이다.

### 2.2.2 하드웨어 컴포넌트

ARM 프로세서는 명령어를 처리하고 데이터를 조작하는 코어가 핵심이다. ARM 코어의 선택은 구현하고자하는 어플리케이션이 원하는 동작과 특성에 맞게 하여야 한다. 시리얼 통신이나 무선 네트워크 등을 지원하기 위한 주변장치가 칩 내부 혹은 외부에 포함한다. 또한 메모리나 인터럽트 지원을 위한 컨트롤러도 주변장치에 속한다. ARM 주변장치는 메모리에 맵핑된 레지스터를 통해 제어한다.

코어와 주변장치를 연결하기 위해 ARM 프로세서의 온칩 아키텍처로 AMBA(Advanced Microcontroller Bus Architecture)를 사용한다[6].

### 2.2.3 소프트웨어 컴포넌트

ARM 시스템은 구동하기 위한 소프트웨어 컴포넌트는 초기화 코드, 운영체제(부트로더 포함), 디바이스 드라이버, 어플리케이션으로 나눌 수 있다. 초기화 코드는 ARM 보드에서 실행되는 첫 번째 소프트웨어로, 특정 아키텍처에 의존적이다. 이 코드는 운영체제로 제어를 넘기기전 수행되는 최소한의 동작을 수행한다. 다음으로 읽혀져 실행될 소프트웨어 코드는 운영체제다[6].

운영체제는 부트로더를 통해 하드웨어 설정 및 레지스터 초기화를 수행한다. 그리고 하드웨어 리소스와 인프라스트럭처를 사용하기 위한 환경을 제공한다[6]. 또한 대부분에 운영체제는 디바이스 드라이버를 포함하여 하드웨어 장치와의 인터페이스를 제공한다. 임베디드 어플리케이션 프로그램은 특정 동작을 수행한다. 운영체제 기반에서 다양한 어플리케이션을 구동 가능하다. RTD-1000A의 구동 프로그램은 선점형 어플리케이션으로 구현된다.

## 3. 기존 RTD-1000 시스템 분석

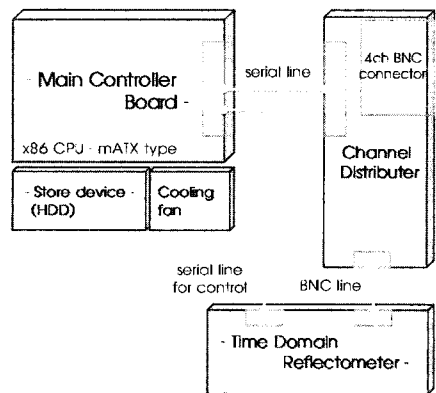
ARM 기반의 RTD-1000A의 구현을 위해서는 기존 RTD-1000의 구성과 기능을 분석할 필요가 있다.

### 3.1 시스템 구성과 기능

RTD-1000A는 기존 RTD-1000을 운영함에 있어 발생한 문제점을 해결하고 성능 향상을 목적으로 가진다. 이를 위해 본 절에서는 기존 시스템의 구성과 기능을 분석한다.

그림 2는 기존 RTD-1000 시스템 구성도이다. 그림에서 채널분배기를 매개로 메인 컨트롤러는 TDR을 제어하고 측정 데이터를 받아 분석 및 처리 후에 모니터링 서버에 전송한다. 기존 RTD-1000의 메인 컨트롤러는 Intel x86 아키텍처 프로세서와 Mini-ITX 규격 메인보드 기반 시스템이다. 또한 수십GB 용량의 하드디스크, SDRAM 메모리카드 그리고 전원공급장치가 설치된 범용 PC의 사양으로 구성되어 있다[4]. 이 구성은 TDR을 제어하기 위한 임베디드 컨트롤러로는 과분한 제원이며, 범용 PC 사양이기 때문에 산업 현장이나 환경이 열악한 조건에서는 그 내구성에 있어 적합하지 않다.

TDR은 메인 컨트롤러에 의해 제어되며 누수감지관의 감지도선을 측정하여 웨이브 데이터를 만든다. 그리고 웨이브 데이터의 분석을 통해 감지도선



(그림 2) 기존 RTD-1000 시스템 구성도

의 단선, 단락을 감지한다.

채널분배기는 감지관의 감지도선과 TDR의 연결부인 BNC Connector를 4채널로 나누고 신호를 분배하는 역할을 한다. 채널분배기는 감지도선을 여러 채널로 나누고 한 시점에 하나의 채널에 신호를 분배하는 역할만을 하는 추가적인 기능이고 RTD-1000 시스템 성능에는 영향이 없으므로 본 논문에서 설계 대상으로 고려하지 않는다.

### 3.2 펌웨어의 구성과 기능

기존 RTD-1000 컨트롤러의 소프트웨어는 크게 운영체제와 펌웨어로 구성된다. 운영체제는 Windows XP Embedded이며 다른 버전과 달리 임베디드 시스템에 탑재할 수 있도록 재구성이 가능하다. 하지만 하드디스크 같은 대용량 보조기억장치에 설치되어야 하고, GUI 기반의 다양한 멀티미디어 유닛을 지원하기 때문에 디스플레이 장치가 필요하지 않는 시스템에서는 리소스의 낭비가 크다는 특징을 갖는다. 더욱이 RTD-1000과 같은 제어 목적을 하는 시스템에 적용할 운영체제로는 적합하지 않다. 그렇기 때문에 본 논문에서 제안한 RTD-1000A의 운영체제는 임베디드 리눅스 커널로 구성하였다. ARM을 지원하는 임베디드 리눅스는 높은 안정성과 우수한 네트워킹을 지원한다는 장점 뿐만 아니라 실시간성 운영체제이며 효율적인 메모리 관리와 전력관리를 가능케 한다[7][8]. 특히 RTD-1000A의 시스템 안정성은 상당히 중요한 사항이며 서버와 같이 항상 운영되어야 하므로 다운되어 제어가 불가능한 상황을 줄이는 것이 매우 중요하다. 그리고 RTD-1000A는 감지관의 상태를 실시간으로 감시하는 역할을 해야 하므로 리얼타임 구동 프로그램이 필수적이다. 이러한 요구에 부합하는 임베디드 리눅스는 시스템 안정성 향상에 기여할 수 있는 보호 메모리와 선제 멀티태스킹이 제공되며, 또한 실시간성을 제공한다.

기존 RTD-1000에 설치된 구동 프로그램은 GUI 기반의 윈도우 응용프로그램으로 TDR과 Modem을

제어하고, 측정된 데이터를 받아서 서버 컴퓨터에 전달하는 역할을 한다. 사용자와 GUI를 통해 시스템 설정이 이루어지므로 RTD-1000 구동 프로그램은 제어 모듈보다 GUI에 많은 리소스가 할당된다. 이는 사용자와의 인터페이스가 항상 요구되지 않는 RTD-1000 시스템으로는 매우 비효율적인 구성이다. 반면 본 논문에서 제안한 RTD-1000A 구동 프로그램은 GUI를 배제하고 구조적이고 직관적인 형태의 최적화된 모듈로 구현하였다. 그리고 시스템 설정 방식은 프로파일 방식을 적용하여 시스템을 전반적으로 효율적이고 간결하게 구성하였다.

## 4. RTD-1000A 설계 및 구현

본 장에서는 3장에서 분석한 기존 RTD-1000의 문제점과 성능향상을 위한 요구사항을 반영한 RTD-1000A 컨트롤러의 설계와 구동프로그램의 구현 방법을 기술한다.

### 4.1 개발 환경 구축 및 요구 분석

현재 ARM 마이크로컨트롤러의 종류는 다양하다. ARM 마이크로컨트롤러는 리소스 낭비를 막기 위해 코어의 버전별 제원을 고려해야 하며 RTD-1000A 구현에 필요한 컨트롤러 디바이스만을 탑재해야 한다. 또한 하드웨어 성능과 비용 사이의 Trade Off를 감안해야 하고, 구동 소프트웨어가 요구하는 최소한의 하드웨어 구성이 마련되어야 한다. 최소한의 하드웨어가 기반 되어야 소프트웨어 개발의 융통성 확보와 비용 문제를 동시에 해결할 수 있다.

RTD-1000A 임베디드 소프트웨어는 초기화 코드를 포함한 펌웨어와 디바이스 드라이버를 포함하는 운영체제 그리고 구동 프로그램으로 구성된다. 펌웨어는 하드웨어를 초기화시키고, 운영체제를 로드하는 역할을 하므로 시스템에서 가장 작으면서 필수적인 프로그램이다. 그리고 펌웨어를 기반으로 구동 프로그램 구현이 가능하다. 하지만 그러한 방

식은 하드웨어 구조에 종속되어 성능 향상을 위한 장치 업그레이드에 제약을 가져올 수 있으며, 프로그램 개발에 어려움으로 기능의 구현과 유지보수가 용이하지 않다[7].

본 논문에서 RTD-1000A의 구동프로그램은 간소화한 리눅스 기반의 프로그램 모듈로 구현된다. 오픈라이센스인 리눅스는 소스가 공개되어 재 컴파일 이 가능하고 다양한 애플리케이션 소스나 디바이스 소스가 공개되어 프로그램 개발이 용이하다. 그리고 리눅스의 아키텍처는 낮은 하드웨어 사양에도 불구하고 높은 안정성과 강력한 네트워크 성능을 가지며 리얼타임 시스템 구현이 가능하다. 그러므로 RTD-1000A 시스템 운영체제로 적합하며 리눅스 커널을 시스템 구동에 필요한 부분을 제외하여 최적화된 구성으로 탑재하여야 한다.

#### 4.2 시스템 구성 및 설계

RTD-1000A의 컨트롤러는 시리얼 방식을 통해 TDR과 통신한다[9]. 그러므로 ARM 마이크로컨트롤러는 USART(Universal Synchronous and Asynchronous serial Receiver and Transmitter) Controller를 탑재하여 구성되어야 한다.

그림 3은 기존 RTD-1000의 문제점을 보완하고 기능을 향상하기 위해 구성한 컨트롤러 보드의 구

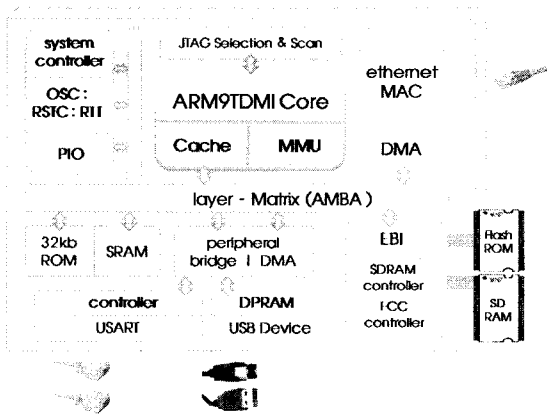
성도이다. RTD-1000A 컨트롤러 보드는 ARM9TDMI 계열 Core를 중심으로 SDRAM 모듈, 플래시 롬 모듈, RS-232 시리얼 포트, USB 포트 그리고 RJ-45 이더넷 포트 등으로 구성된다. 다음 표 1은 기존 RTD-1000의 컨트롤러와 RTD-1000A의 ARM 컨트롤러의 사양을 비교한 것이다.

(표 1) 컨트롤러 보드 사양 비교

	RTD-1000A	기존 RTD-1000
프로세서	ARM926EJ-S(200Mhz)	Pentium III(800Mhz)
메인 메모리	64Mbyte	256Mbyte
주 저장장치	NANDFlash(256MB)	HDD(40GB)
지원 포트	RS-232, RJ-45, USB	D-SUB, RS-232, RJ-45, USB, EIDE, audio, etc...
크기(mm)	100×170×15	170×170×55
요구전원(전력)	5V DC(10 Watt)	5V&12V DC(250 Watt)
내부 온도	약 19°C	약 33°C(cooling fan 미 장착시)

표 1을 보면 RTD-1000A는 기존 RTD-1000에 비해 메인메모리와 주 저장장치의 용량을 최소화하였고, 쿨링팬과 불필요한 포트를 제거하여 컨트롤러 보드의 크기를 크게 줄인 것을 확인할 수 있다. 이는 전체적인 하드웨어 리소스가 75%이상 감소되어 짐을 보여준다. 또한 내구성에 문제되어지는 하드웨어 부품도 제거할 수 있어 커다란 성능 향상과 함께 시스템 안정성도 보장된다.

컨트롤러의 ARM 코어는 ARM 9로 선택하였다. ARM 9의 Clock Speed는 약 200MHz 정도로 RTD-1000A 구동에 무리가 없고 운영체제를 사용하기 위한 MMU(Memory Management Unit)도 탑재되어있다[10]. 코어는 버스를 통해 별도의 32KB 크기 SRAM, 외부 메모리 컨트롤러와 인터페이스, Ethernet MAC 컨트롤러 그리고 Peripheral DMA와 Bridge가 연결된다. 이렇게 구성된 ARM 컨트롤러는 기존의 RTD-1000의 범용 PC사양의 시스템을 대체하여 리소스 낭비를 줄이고 성능을 높임과 동시에 기존 시스템에서 발생한 문제점을 해결할 수 있다.



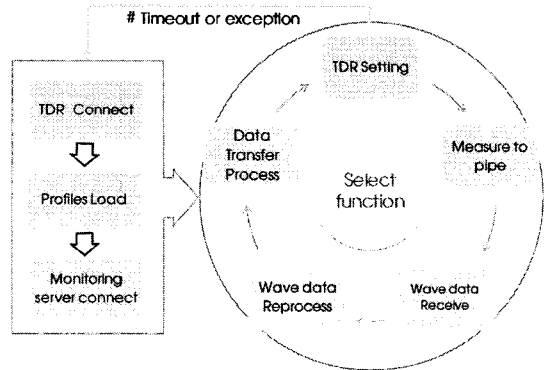
(그림 3) RTD-1000A 컨트롤러 보드 구성도

### 4.3 구동 프로그램 구현

앞 절에서 RTD-1000A를 구현하기 위한 시스템 개발환경 구축과 ARM 기반 RTD-1000A의 컨트롤러 보드를 설계하였다. 이 RTD-1000A 시스템을 구동하기 위해서는 ARM에서 실행 가능한 소프트웨어 모듈이 필요하다. RTD-1000A의 리눅스 기반의 프로그램 코드는 기존 RTD-1000의 윈도우 프로그램과 달리 구조적이고 직관적이어야 한다. 이는 다소 제한적인 리소스를 가진 ARM 시스템에 적합하도록 간략하고 명확한 프로그램을 만들기 위함이다.

RTD-1000A의 구동 프로그램은 리눅스 기반의 어플리케이션 프로그램이며 기능에 따라 크게 TDR을 제어하여 측정된 데이터를 얻는 측정모듈과 측정된 데이터를 서버로 전송하는 전송모듈로 나눈다. 측정 모듈의 수행 절차는 측정을 준비하기 위해 TDR을 세팅하고 측정명령을 전송한다. 측정이 완료되면 TDR로부터 웨이브 데이터를 받아 필요한 정보를 삽입하고 재배치하는 과정을 거쳐 서버 모니터링 프로그램이 인식할 수 있는 데이터 타입으로 만든다. 전송모듈은 서버에 FTP(File Transfer Protocol)을 이용하여 데이터를 전송한다. 또한 모니터링 프로그램 서버에 접속 상태, 데이터 전송의 무결성을 감시하는 기능을 포함한다. 두 모듈은 일련의 절차에 의해 주기적으로 수행되어야 한다. 이 절차 중 한 단계라도 처리되지 않으면 정확한 측정이 되지 못한다. 이를 방지하고 구동 프로그램이 정확한 측정을 하기 위해 select 모델을 적용하였다.

RTD-1000A 서브시스템인 TDR의 제어는 직렬통신을 이용한 데이터 교환 방식으로 수행한다. TDR 제어를 통해서 정확한 측정을 하기 위해서는 철저한 동기화 과정을 적용하여야 한다. 이에 RTD-1000A의 구동 프로그램은 select 모델을 사용하여 모든 과정을 동기화하였다. 핵심적인 역할을 하는 select 함수는 TDR과의 연결 상태를 감지하여 다음 작업을 결정한다. select 함수는 커널에 여러 개의 이벤트들 중 하나에 대해 대기하게 하고 해당



(그림 4) 구동프로그램의 select 모델

이벤트가 발생하거나 지정한 시간이 경과하면 프로세서에게 알리는 역할을 한다. 본 논문에서 구현한 구동프로그램은 이 두 가지 특징을 이용하는데, TDR에 측정 주기 및 간격을 조절하게 하고 측정 과정에서의 timeout과 exception으로 인해 발생하는 오류를 차단해준다. select 함수의 주요 매개변수는 file descriptor의 최대개수, I/O field, timeout struct이다. file descriptor는 TDR과의 연결 소켓을 의미하며 timeout struct는 상태를 감지하는 제한시간을 의미한다.

그림 4는 select 모델을 적용한 RTD-1000A 구동 프로그램의 작업을 순차적으로 수행하는 것을 보여준다. 또한 에러 발생 시 수행하던 작업을 반복하도록 하여 작업의 누락을 방지하고, TDR과 동기화하여 제어 명령어가 중복 전송되는 것을 막는다. select 모델은 단순히 TDR에 전송할 명령어를 가지는 것이 아니라 세부적인 동작이 기술된 함수의 엔트리 포인터를 가진다. 이는 select 함수를 TDR과의 통신에 이용할 supervisor로만 사용되는 것이 아니라 RTD-1000A 구동 프로그램의 모든 동작을 select 함수 안에서 이루어지게 함으로써 TDR에서 측정된 데이터의 훼손을 미연에 방지하고 정확한 측정 데이터를 서버의 모니터링 프로그램에 전달할 수 있도록 하기 위함이다. 그리고 시스템의 안정성을 높이기 위해 측정 횟수를 체크하여 일정한 주기마다 시스템을 초기화하는 기능을 루틴에 추가하였다.

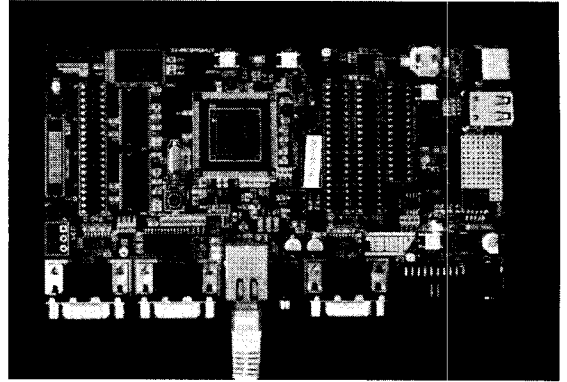
본 구동프로그램은 직관적이며 가시성이 뛰어나므로 프로그램의 업그레이드 및 변동사항을 적용하기가 매우 용이하다. 그렇지만 전송모듈에 적용할 FTP 프로그램은 코드가 다소 복잡하여 본 구동프로그램 내에 같이 구현할 경우 프로그램 전체를 복잡하게 하여 유연성을 해친다. 그러므로 본 구동프로그램의 전송모듈의 FTP는 공유라이브러리 방식으로 구현하였다. 공유라이브러리는 라이브러리를 업데이트해도 본 프로그램은 변경 없이 그대로 사용할 수 있도록 지원한다. 이는 select 모델 안에 측정모듈과 전송모듈을 동작하게 하면서 기능 업그레이드나 유지보수 시에는 서로 영향을 주지 않아 프로그램의 유연성을 상당히 높이고 전송모듈을 다른 프로토콜 방식으로 쉽게 교체할 수 있는 이점을 가진다.

RTD-1000A 컨트롤러는 다양한 임베디드 프로세서와 운영체제 중에서 ARM과 리눅스 조합을 선택하였다. 이 조합은 현재 임베디드 시스템에서 가장 보편적이며 지원 소프트웨어가 다양하다. 그러므로 다른 임베디드 프로세서나 운영체제에 비해 RTD-1000A의 구동 프로그램의 구현 및 개량에 매우 효과적이다.

#### 4.4 시뮬레이션 및 성능 비교

본 논문에서 RTD-1000A 프로토타입을 구현하고 기존 RTD-1000과의 성능비교 및 검증을 하기 위해 누수감지관 가상 네트워크를 설정하고 시뮬레이션하였다. 테스트베드는 채널분배기를 제외하고 지상에서 1채널의 감지관과 연결하는 것으로 하였고, 모니터링 서버와의 네트워크를 위해 TCP/IP 프로토콜로 LAN을 통해 인터넷망에 접속하였다.

그림 5는 ARM 마이크로컨트롤러 기반의 RTD-1000A 메인 컨트롤러 보드의 모습이다. 기존 RTD-1000 컨트롤러 보드 대비 30%대로 소형화 되었으며 하드웨어 구조가 복잡하지 않으므로 내구성 면에서도 뛰어나다. 또한 컨트롤러 구성하는데 외부 저장장치와 같은 별도의 주변기기가 요구되지 않



(그림 5) RTD-1000A 메인컨트롤러 보드

는다.

다음 표 2에서는 RTD-1000A와 기존 RTD-1000 구동 프로그램의 성능 비교를 위해 다음과 같이 측정시간, 프로그램 파일 크기, 설정 방식, 메모리 사용량을 비교하였다. 비교 데이터의 타당성을 높이기 위해 구동 프로그램 이외의 성능에 영향을 주는 외부요인은 최대한 제거하였고 측정시간 비교 데이터는 10회 측정을 반복하여 얻은 평균값이다.

(표 2) 구동 프로그램 성능 비교

	RTD-1000A	기존 RTD-1000
측정시간	평균 19초	평균 63초
파일크기	41KB	9.4MB
설정방식	Profiles	GUI
점유 메모리	최대 2000 KB	최대 20,000 KB

측정 시간 비교는 1채널을 기준으로 동일한 길이를 측정된 결과로 측정 명령을 시작으로 웨이브 데이터의 전송 완료까지를 체크하였다. 정확한 시간을 위해 time 함수를 이용하였다. 비교 결과 RTD-1000A의 구동프로그램이 평균 65%이상 향상된 응답 성능을 보였다[11].

구동 프로그램의 파일 크기 비교는 공유 라이브러리를 포함한 전체 파일크기로 NTFS(New Technology File System)상에서 측정하였다. 기존 RTD-1000의 프로그램 크기가 약 10MB인 반면

```
[root@localhost rtd-1000a]# ./rtd-1000a
#####
## RTD-1000A Operation Program For Linux(Ver 1.0) ##
## Copyright by Embedded System Lab, SCH Univ. ##
#####
* Setting on RTD-1000a ...
* Serial port opened of /dev/ttyS0(fd:3)...
* Connect to node tdr002-1@fid.sch.ac.kr
>> RTD-1000A Start at Mon Mar 3 11:38:23 2008
[0]Measure =====> Complete.
[1]Measure =====> Complete.
[2]Measure =====> Complete.
[3]Measure =====> Complete.
[4]Measure =====> █
```

(그림 6) RTD-1000A의 구동

RTD-1000A의 구동 프로그램의 크기는 기능을 확장한다고 해도 100KB 미만이 될 것이다. 설정 방식도 기존 RTD-1000의 경우 GUI 방식으로 리소스 낭비가 크고 별도의 디스플레이 장치가 필요하다. 반면 RTD-1000A의 경우 프로파일 방식으로 호스트 시스템에서 작성하여 구동 프로그램에 손쉽게 로드하여 적용할 수 있다.

RTD-1000A는 시뮬레이션에 적합하게 터미널을 구성하였다. 그림 6은 위와 같은 테스트베드에서 RTD-1000A가 시뮬레이션 되고 있는 터미널을 보여주고 있다. RTD-1000A 시스템의 안정성 테스트를 위해 30일 동안 계속해서 시뮬레이션 하였고 터미널을 통해 이를 수시로 확인하였다. 그 결과 인터넷망 이상으로 몇 건의 웨이브 데이터 전송 실패를 제외하고는 안정성 있게 동작하는 것을 확인할 수 있었다.

위와 같은 시뮬레이션을 통해 본 논문에서 제안한 RTD-1000A는 기존 RTD-1000에 비해 응답성, 효율성 그리고 안정성 면에서 우수한 성능을 보였으며, 서버의 GIS 기반 모니터링 프로그램에서 요구하는 모든 데이터 요구를 충족시켰다고 판단된다.

## 5. 결론

본 논문에서는 기존 RTD-1000 시스템을 운영함에 있어 발생한 문제점을 확인하고 이를 해결하기

위한 방안으로 ARM 마이크로컨트롤러를 이용한 RTD-1000A의 컨트롤러 구성과 개발을 위한 최적의 방법을 제안하였다. 기존 RTD-1000의 범용 PC 사양 시스템을 ARM 시스템으로 대체할 경우 최적화된 구성으로 리소스 낭비를 줄이고 시스템 구축비용을 절감할 수 있으며 내구성 역시 획기적으로 향상 가능하다. 또한 유지 보수가 매우 용이할 것으로 판단된다. 그리고 제안한 구동 프로그램 사용으로 기존 RTD-1000의 성능을 한층 업그레이드 시킬 수 있으며 시스템의 안정성이 뛰어나 항시 구동되어야 하는 시스템으로서 역할을 시스템 중단 없이 충분히 감당할 수 있을 것으로 예상된다. 향후 연구 과제는 현장에서 좀 더 나은 성능과 정확성 그리고 안정성을 수반하는 RTD-1000A의 현장 적용 및 상용화이다.

## 참고 문헌

- [1] 김민호, 홍인식, ARM 시스템을 이용한 RTD-1000 SoC에 관한 연구, '07 추계학술발표대회논문집, 한국멀티미디어학회, pp.623~626, 2007년 11월
- [2] 강병모, 홍인식, GIS상에서 TDR을 사용한 누수감지관과 모니터링 시스템에 관한 연구, Journal of Korea Multimedia Society Vol 7, No. 4, 한국멀티미디어학회, 2004년 4월
- [3] 강병모, 홍인식, 원격 상수도관망 누수감지 시스템에 관한 연구, 정보처리학회논문지, Vol. 11, No. D, 한국정보처리학회, pp.1311~1318, 2004년
- [4] (주)WACON, www.wacon.co.kr
- [5] ARM Ltd, www.arm.com
- [6] Andrew N Sloss, Dominic Symes, Chris Wright, "ARM System Developer's Guide", 사이텍미디어, Printed in Korea, 2004
- [7] IBM, http://www-903.ibm.com/kr/linux/
- [8] DENX, www.denx.de/wiki/UBoot
- [9] (주)Nanotronix, www.nano-tronix.com



[10] 김형태, ARM Processor의 특성 및 최신동향, 전력전자학회지 제13권 제1호 PP. 20~25, 전력전자학회, 2008. 2

[11] 황요섭, 안성용, 심재홍, 이정아, ARM용 내장형 소프트웨어의 정적인 수행시간 분석도구, 정보과학회논문지:컴퓨팅의 실제 제11권 제1호 pp.15~25, 한국정보과학회, 2005.2

## ● 저 자 소개 ●



### 김 민 호

2007년 순천향대학교 정보기술공학부(공학사)  
2007~현재 순천향대학교 대학원 컴퓨터학과(석사과정)  
관심분야 : 임베디드 시스템, USN, RFID  
E-mail : mindhawk@sch.ac.kr



### 홍 인 식(교신저자)

1981년 한양대학교 전자공학과(공학사)  
1986년 한양대학교 대학원 전자공학과(공학석사)  
1988년 한양대학교 대학원 전자학과(공학박사)  
1991~1995년 순천향대학교 공과대학 전산학과 전임강사  
1995~1999년 순천향대학교 공과대학 컴퓨터학부 조교수  
1999~2004년 순천향대학교 공과대학 정보기술공학부 부교수  
2004~2005년 순천향대학교 공과대학 정보기술공학부 교수  
2005~현재 순천향대학교 공과대학 컴퓨터학부 교수  
관심분야 : 유비쿼터스 컴퓨팅, 임베디드 시스템, 유무선네트워크  
E-mail : ishong@sch.ac.kr