

# 도시형 지진방재시스템을 위한 네트워크 구성, 시간관리 및 데이터 저장 방법

## Network Configuration, Time Management, and Data Storage for Urban Earthquake Disaster Preventing System

최 훈<sup>†</sup> · 윤 주 상<sup>\*</sup> · 허 경 용<sup>\*\*</sup>  
(Hun Choi · Joosang Youn · Gyeongyong Heo)

**Abstract** - In this paper, we propose a precise time management and time synchronization based on real-time data storage and transmission scheme in design of seismic data acquisition system for urban earthquake disaster preventing system (UEDPS). It is possible to improve the performance of the existing research results through the proposed methods. To evaluate the performances of the proposed methods, we implemented a prototype system(H/W & S/W) and performed some experiments with real seismic data and test equipment generated data as the input.

**Key Words** : Earthquake, Disaster preventing system, Real time clock, Ethernet, Seismic data management

### 1. 서 론

지진조기경보시스템(EEWS; Earthquake Early Warning System)은 지진발생으로 인한 직·간접적인 피해를 줄이기 위한 관측 및 분석 그리고 제어기술을 총망라하는 통합시스템이다[1-5]. 세계 각국은 지진 재해 경감을 위해 국가 지진 관측망을 활용하여 전 국토를 경보대상으로 하는 국가지진조기경보시스템(NEEWS; National EEWS)을 개발 중이거나 운영하고 있다. 지진 피해 사례를 보면 지진발생 시 그 피해가 국가의 존립을 위협할 만큼 크며 일부 지역에 국한하지 않고 광범위하게 나타나게 되므로 NEEWS는 국가 차원에서 반드시 확보해야하는 기술이다. 이러한 NEEWS의 효율을 극대화하기 위해서는 지진데이터 수집에 필요한 고밀도 기관 관측망 구축이 선행되어야 하지만 관측망 확대를 위해서는 고가의 지진관측 및 분석장비 그리고 운영 인력확보에 막대한 비용이 소요되는 단점이 있다. 이러한 문제점을 해결하기 위해 저비용으로 NEEWS의 관측망 밀집도를 높일 수 있는 미국 Caltech의 CSN(Community Seismic Network), 대만 Sanlien의 Seismic Switch, Palert, 일본 Cygnet사의 直下く(직하지진 검출기) 등의 저해상도 네트워크 지진검출시스템 개발이 활발히 진행되고 있다. 네트워크

지진검출시스템은 고성능 지진관측용 센서를 저비용 저해상도 MEMS(micro electro mechanical systems) 센서로 대체하고 관측용 장비보다 상대적으로 저렴한 임베디드 시스템 형태이므로 NEEWS에 비해 구축비용을 줄일 수 있다. 그러나 MEMS 가속도 센서는 시스템 비용을 현저히 줄일 수 있지만 고가의 지진관측 전용 센서에 비해 센서가 갖는 고유한 특성에 의해 부가잡음(additive noise) 수준이 높은 단점이 있다. 최근 국내 연구팀에 의해 EEWS의 관측망 확장 및 생활지역 내 지진 재해 경감을 위한 도시형 지진방재시스템(UEDPS: Urban Earthquake Disaster Preventing System) 개발을 위한 다양한 연구가 시도되었다[6-8]. [6]은 보급형 MEMS 가속도 센서를 활용한 FPGA-ARM기반의 지진 데이터 취득 시스템(SDAS: Seismic Data Acquisition System) 설계 및 구현 방법을 제시하였다. [7]에서는 시간-주파수 누적 변화량(ATFC: Accumulated Time-Frequency Changes)을 기반으로 하는 지진 이벤트 자동 검출 알고리즘, 그리고 [8]에서는 다수 개의 센서의 중합처리 기법을 활용하는 배경잡음 제거방법을 제시하고 효과를 검증하였다. 지진 데이터 측정 시스템에서 GPS 시각과 각 축(EW, NS, UD)별 측정 데이터의 취득 시각의 동기화는 시스템의 데이터 취득 성능뿐만 아니라 취득 후 분석 및 처리 알고리즘의 성능을 결정하는 주요 요소이다. 특히, 다수 개의 센서를 활용할 경우 보다 정밀한 시각 동기를 위한 시간 관리가 요구된다. 또한 다수의 위치에서 측정된 지진 데이터를 데이터 관리용 서버 시스템으로 전송 및 저장할 때 시각 동기화는 매우 중요하다.

본 논문에서는 도시형 지진방재시스템(UEDPS)을 위한 다수 개 MEMS 센서기반의 개선된 지진데이터 취득시스템의 설계 및 구현에 요구되는 정밀한 시각 관리와 시각 동기화에 기반한 실시간 데이터 저장 및 전송 방안을 제안한다.

<sup>†</sup> Corresponding Author : Dept. of Electronic Engineering, Dong-Eui University, Korea

E-mail: hchoi@deu.ac.kr

<sup>\*</sup> Dept. of Multimedia Engineering, Dong-Eui University, Korea.

<sup>\*\*</sup> Dept. of Electronic Engineering, Dong-Eui University, Korea

Received : October 29, 2014; Accepted : November 14, 2014

제안한 방법은 기존 연구결과[6-8]를 기초로 하여 기존 하드웨어[6]의 개선 방안과 개선된 하드웨어를 위한 펌웨어 설계 방안을 제시함으로써 지진방재시스템의 지진 데이터 취득 성능의 개선과 이더넷을 사용한 안정한 실시간 데이터 전송 및 효율적 관리를 가능하게 한다. 성능 평가를 위해 제안한 방법을 적용한 프로토타입 시스템을 제작하고 가진기(실 지진과와 동일한 진동 발생 장치)를 사용한 실험을 수행하여 유효성을 검증하였다.

## 2. 지진데이터 취득 시스템

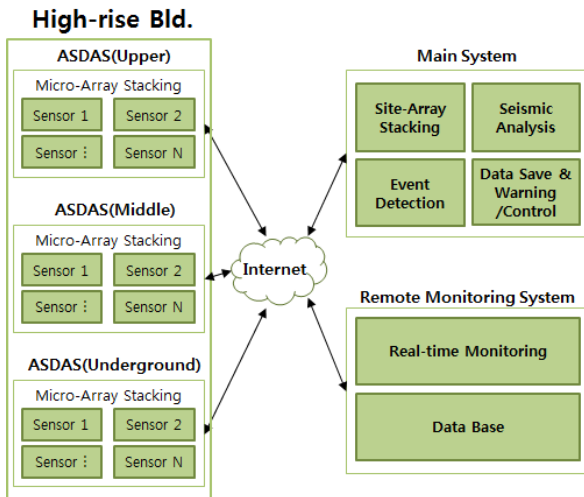


그림 1 도시형 지진방재용 전체 시스템 구성  
 Fig. 1 Total system configuration for an urban earthquake disaster prevention

치에 설치된 ASDAS의 블록도를 그림 2에 나타내고 있다. 제안한 ASDAS는 [6]에서 제안한 시스템을 기본으로 개선된 사양으로 설계되었다. ASDAS는 다수 개의 14비트 디지털 출력을 갖는 MEMS 가속도 센서(Freescale사의 MMA8451Q)와 FPGA(Cyclone II) 및 ARM 프로세서(Cortex M3)로 구성되며 Wiznet사의 W5200 칩을 적용함으로써 이더넷을 통한 측정데이터의 전송이 가능하도록 하였다. 또한 NAND Flash 메모리와 Micro SD 카드를 사용하여 측정된 데이터의 저장 및 통신 로그 기록이 가능하도록 하였다.

## 3. 지진데이터 취득시스템 개선 방안

### 3.1 정밀한 시각 관리 방안

지진 데이터 취득 시스템에서는 정밀한 시각 관리가 중요한 요소 중 하나이므로 이를 위해 RTC(Real Time Clock) 기능을 ASDAS의 설계에 반영하였다. 사용한 STM사의 ARM 프로세서에 내장되어 있는 온칩 RTC는 ASDAS의 시스템 시간 정보를 비롯하여 MEMS 센서의 샘플링 주기, 센서 각 축의 취득 시각, 데이터 전송 또는 전송 실패 시의 시각 등 실시간 시각 정보를 정밀하게 제공한다. 특히, 정밀한 샘플링 주파수(100Hz)로 각 축의 데이터를 수집하고 전/후처리 및 분석과 동시에 Micro SD 카드에 저장해야 하며 인터넷 연결이 비정상적으로 해제되었을 경우 곧바로 NAND Flash 메모리에 임시 데이터를 저장해야 하므로 이 때 RTC 기능은 매우 중요한 역할을 하게 된다. 또한 RTC는 ASDAS의 시스템 자원의 효율적 배분을 위해 타이머 모듈을 사용하여 각각 다른 타이밍을 가지고 주기적으로 시간 자원을 배분하는데 사용된다.

### Advanced SDAS

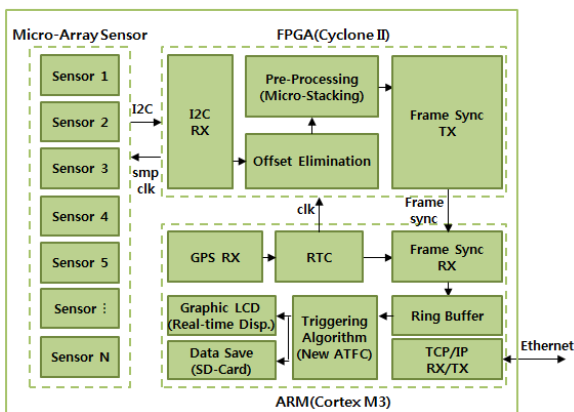


그림 2 미소배열 센서를 사용하는 개선된 지진 데이터 취득 시스템 블록도  
 Fig. 2 System block diagram of advanced Seismic Data Acquisition System with Micro-Array Sensors

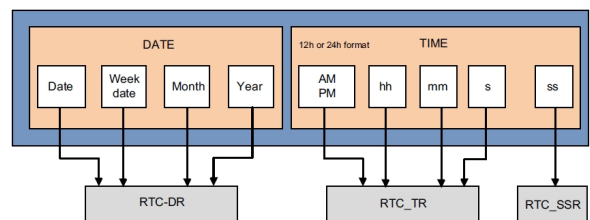


그림 3 RTC의 레지스터[9]  
 Fig. 3 Registers of RTC[9]

그림 1은 지진 데이터 취득 시스템을 고층 빌딩의 상층, 중층, 그리고 하층에 각각 설치하고 빌딩 통제실에 위치한 주 시스템 및 원격 모니터링 시스템과 이더넷(ethernet)으로 연결된 UEDPS의 전체 구성도를 나타내고 있다. 각 각의 위

그림 3에서 DATE 영역 모든 레지스터(register)들과 TIME 영역의 ss(Sub Second) 레지스터를 제외한 RTC\_TR 레지스터가 현재 시간을 관리해주는 레지스터이다[9]. RTC는 외부에서 클럭(clock)을 입력받아서 동작하며 ASDAS에서는 32.768kHz 수정 진동자(크리스탈)를 사용하였다. 로드 캐패시턴스(Load Capacitance)는 범용적인 사용법에 따라 5~8pF 사이의 용량으로 사용하였다. 보다 정확한 기준 클럭을 만들기 위해 RTC는 비동기 Prescaler PREDIV\_A와 동기 Prescaler PREDIV\_S를 통해 클럭을 분주하였다. 그에 대한 내용은 아래 그림 4와 같다. ASDAS에서 PREDIV\_A는 127 (div128) / PREDIV\_S 255 (div256)을 적용하여 정밀한 1Hz 기준 클럭을 생성하였다.

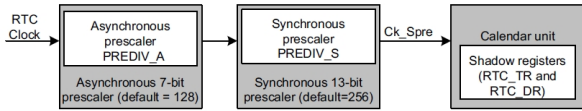


그림 4 RTC Clock의 PREDIV\_A와 PREDIV\_S를 통한 클럭 분주[9]

Fig. 4 Clock frequency division over PREDIV\_A and PREDIV\_S the RTC Clock[9]

그림 5는 펌웨어 상에서 RTC를 설정하는 부분 중 LSE (Low Speed External clock = 크리스탈 32.768KHz)를 활성화(ENABLE)시키는 루틴이다. 하드웨어 구성 중에 로드 커패시턴스가 8pF을 넘어서는 경우 while (RCC\_GetFlagStatus) 구문에서 루프가 걸리게 된다. RTC의 안정적인 하드웨어 구성을 위해서 5~8pF의 하드웨어 구성을 따르는 것이 좋다.

```

/* Reset Backup Domain */
BKP_DeInit();

/* Enable LSE */
RCC_LSEConfig(RCC_LSE_ON);
/* Wait till LSE is ready */
while (RCC_GetFlagStatus(RCC_FLAG_LSERDY) == RESET);

/* Select LSE as RTC Clock Source */
RCC_RTCLKConfig(RCC_RTCLKSource_LSE);
    
```

그림 5 펌웨어 상의 LSE 사용 설정  
Fig. 5 Using LSE on firmware settings

또한 설계한 ASDAS에서는 정확히 1초 간격으로 이벤트를 발생시키는 것이 필요한데 이를 위해서 RTC의 WakeUp 이벤트[9]를 사용하여 IT\_SEC 인터럽트를 생성시키는 방식으로 이를 구현하였다. WakeUp 이벤트 생성 순서는 먼저 wakeup 타이머를 설정하고 클럭을 입력한다. 이후 wakeup 타이머에 프로그램 값을 입력하고 wakeup 이벤트를 재설정하는 방식이다. ASDAS에서는 초기 부팅 시 또는 재 부팅 시 GPS로부터 수신된 시각정보를 사용하여 시스템 시각을 설정한다. GPS는 직렬통신 방식으로 통신하며 ASDAS에서는 수신된 GPS 정보를 파싱해 시각 설정에 필요한 데이터만 뽑아낸다. GPS 데이터 구분분석(parsing)에 대한 규칙과 소스 코드의 일부를 그림 6과 그림 7에 각각 보였다.

그림 7에서 ch는 시리얼 데이터의 한 바이트(Byte)를 나타내며 '\$GPSRV'와 같은 문자가 계속해서 들어오게 된다. 이 중에 시스템에서는 아스키 코드 ';'를 계속 조사하여 특정 횟수가 넘으면 시간의 데이터가 있다고 판별하고 시각 데이터를 추출해서 가져오도록 하였다. 또한 수신된 GPS 데이

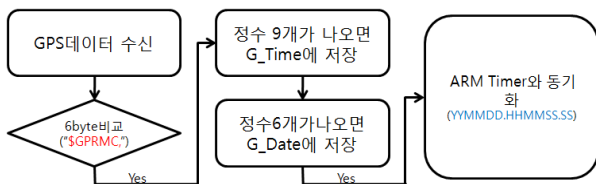


그림 6 GPS 데이터 파싱 규칙도  
Fig. 6 Block diagram for parsing of GPS Data

```

if(46 <= ch && ch <= 57) //
{
    g_tempTime[g_count] = ch;
    g_count++;
}

if(g_count == 6) // 데이터 출력 (DD.MM.YY)
{
    for(int i = 0; i < 6; i++)
        g_date[i] = g_tempDate[i];
}
    
```

그림 7 GPS 파싱 소스코드의 일부  
Fig. 7 Some of the GPS parsing source code

터에는 날짜도 포함되므로 이를 추출하기 위해 초 데이터를 계속 조사하다가 일정 횟수 (코드에서는 g\_count)를 넘기게 되면 날짜로 인식하고 그 데이터를 g\_date 배열에 계속해서 파싱하고 파싱된 데이터는 RTC의 날짜 변수에 곧바로 할당이 되어 Backup 레지스터의 데이터가 교체가 된다. BKP 레지스터 (BackUp Register)는 ASDAS에서 중요한 역할을 한다. STM32 계열의 ARM 코어 기반의 칩들은 RTC 기능과 더불어 BKP 레지스터를 지원한다. 그림 8에서 RTC 모듈과 함께 Backup reg이 backup register를 담당하는 부분이다. RTC와 Backup 레지스터는 Vbat 핀에 공급되는 전원으로 운영되며 backup 레지스터는 42개의 16비트 레지스터로 84 바이트의 사용자 애플리케이션 데이터를 Vdd 공급이 없을 때에도 사용할 수 있다. 이는 ASDAS에서 파라미터 변경이 라던지 F1 시리즈에는 제공하지 않는 날짜를 기록하고 가져오기 위해 필수적인 기능이다. ASDAS에서 backup 레지스터를 사용한 규칙은 다음의 그림 8에 나타내었다.

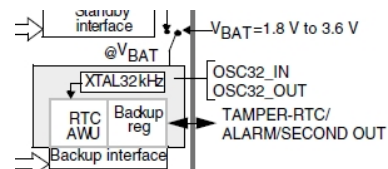


그림 8 Backup 레지스터 규칙  
Fig. 8 Rule of Backup Register

그림 8에 나타낸 규칙으로 데이터를 저장하였으며 DR4 ~ DR16까지는 윈도우즈 애플리케이션과 연동하여 변경이 요청된 때에는 파싱을 통해 각 데이터를 기록하도록 하였다. 이렇게 Backup 레지스터에 기록된 데이터는 타겟 보드가 재부팅을 하는 경우에도 삭제나 변경이 되지 않으므로 재부팅을 하고 Backup Register에 기록이 있는 경우는 그 값을 시스템의 파라미터에 적용시켜 부팅하는 방식으로 펌웨어를 구성하였다.

### 3.2 안전한 데이터 전송을 위한 방안

국가 지진 관측망을 사용하여 전 국토를 지진 방재 대상으로 하는 국가조기경보시스템과 유사하게 도심의 생활지역 내 특정 지역을 대상으로 하는 UEDPS는 지진 데이터 측정시스템인 로컬용 ASDAS, 주 시스템인 서버용 ASDAS, 그리고 원격 모니터링 시스템으로 구성된다. 이때 각각의 로컬용

ASDAS와 서버용 ASDAS(주 시스템)는 네트워크를 통해 연결되어 실시간으로 데이터 전송이 이루어진다. 제한한 ASDAS의 이더넷 통신모듈의 구현을 위해 SPI(Serial Peripheral Interface) 성능을 향상시킨 Wiznet社의 W5200을 사용하여 설계하였다. 하나의 칩 내에 TCP/IP 프로토콜 처리 및 10/100 Ethernet PHY와 MAC을 구현한 W5200은 인터넷 연결을 보다 손쉽게 구현할 수 있는 장점이 있다. 제한한 UEDPS는 동일한 하드웨어 사양으로 설계된 ASDAS의 펌웨어만 달리하여 주 시스템(server)과 로컬 지진데이터 취득시스템(client)을 구성함으로써 효율적인 구축이 가능하다. 그림 9는 고층 건물에 구축한 지진방재시스템 시나리오를 보이고 있다.

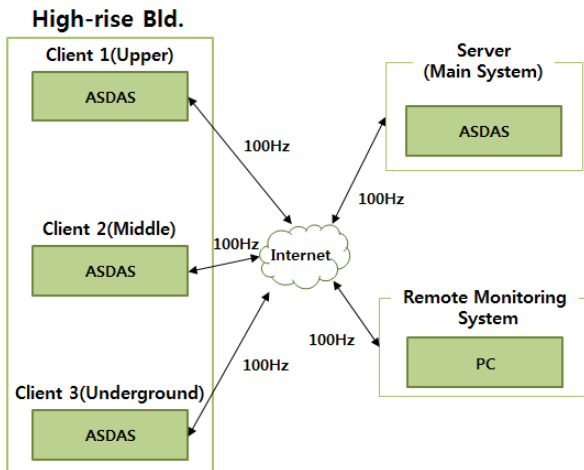


그림 9 고층건물 내 구축된 지진방재시스템 구성도  
 Fig. 9 System configuration of UEDPS in a high-rise building

그림 9에서 3대의 로컬용 ASDAS는 건물의 상층, 중층, 하층에 각각 설치되고 각 층에서 취득된 100Hz의 지진 데이터는 인터넷을 통해 서버용 ASDAS(주 시스템)로 실시간 전송된다. 또한 로컬용 ASDAS에서 측정된 지진 데이터와 서버용 ASDAS에서 수집된 지진 데이터 및 분석·처리된 결과는 외부 인터넷 망이나 로컬 네트워크에 접속된 모니터링 시스템의 사용자 프로그램을 사용하여 확인 및 저장이 가능하다.

각 ASDAS에서 인터넷 물리적 규격 하에 사용된 프로토콜은 TCP/IP 방식을 사용하였다. 이처럼 연결지향적 프로토콜을 사용한 이유는 데이터의 신뢰도나 안정성이 중요한 계측 용도의 UEDPS에서 UDP나 그 밖의 사용자 임의의 프로토콜은 취약하기 때문이다. ASDAS의 펌웨어 구현에서는 클라이언트가 서버와 접속한 이후 서버는 접속한 클라이언트를 소켓과 아이피 주소로 구분하여 어떤 클라이언트가 접속을 하였는지 확인한 이후 데이터를 보내도록 하였다. 또한 네트워크 연결 상태의 예외적인 상황에 대비하여 몇 가지 대비책을 마련하였다.

첫 번째 방법에서는 서버용 ASDAS에서 소켓의 현재 상태를 상시 확인한다. 서버측 소켓은 어떤 로컬용 ASDAS가 접속하였는지 구분하는 용도 외 현재 네트워크의 상태를 나타내는 기준으로 사용된다. 그림 10은 네트워크 연결 안정성 확보를 위해 사용한 소켓상태 확인 방법이다.

서버용 ASDAS에서 소켓은 연결대기 상태일 때 SOCKET\_INIT 상태이고 로컬용 ASDAS가 접속이 되면

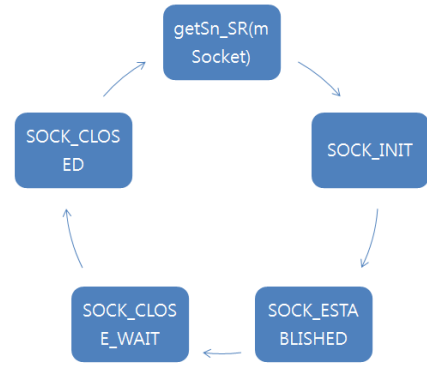


그림 10 소켓상태 확인 방법  
 Fig. 10 Checking method of socket status

SOCKET\_ESTABLISHED 상태로 바뀌게 된다. 펌웨어 상에서 소켓 상태를 확인하는 구문은 그림 11과 같다.

```
switch (getSn_SR(mSocket))
{
    case SOCKET_ESTABLISHED:
    case SOCKET_CLOSE_WAIT:
    case SOCKET_INIT:
    case SOCKET_CLOSED:
}
```

그림 11 소켓상태 체크를 위한 소스코드  
 Fig. 11 Source code for checking the socket status

소켓의 현재 상태는 getSn\_SR(SOCKET) 메서드를 통해 구할 수 있고 펌웨어 상에서 소켓의 상태의 확인 및 처리는 SOCKET\_ESTABLISHED, SOCKET\_CLOSE\_WAIT, SOCKET\_INIT, SOCKET\_CLOSED 4가지 경우로 나뉘어 판별하게 된다. 소켓이 연결 상태일 때에도 전송 데이터의 안정성과 네트워크 상태를 확인하기 위해 SOCKET\_ESTABLISHED 일 때만 데이터를 전송하도록 하였다. 네트워크가 예상치 못한 상황으로 연결이 해제되었을 때에는 소켓 상태가 SOCKET\_CLOSE\_WAIT가 되는데 이 상태가 되면 네트워크가 비정상적으로 종료되었다고 판단할 수 있다. 이 때 펌웨어 상에서는 아래 그림 12에서와 같이 연결을 종료하는 disconnect 메서드와 모든 플래그를 클리어(clear) 한다.

두 번째 방법에서는 정상적으로 종료하지 못한 데이터를 담

```
case SOCKET_CLOSE_WAIT:
    printf("\r\nSocket No. %d : CLOSE_WAIT", mSocket);

    // disconnect and set status
    disconnect(mSocket);
    ch_status[mSocket] = 0;

    E1Flag = false; // Reset E1 flag
    E2Flag = false; // Reset E2 flag
    PCFlag = false; // Reset PC flag
    break;
```

그림 12 네트워크의 비정상적 종료 상황에서의 처리 방법  
 Fig. 12 Processing method in the abnormal condition of the network termination

프(dump) 데이터로 규정하고 로컬용 ASDAS 상의 NAND 메모리(버퍼)에 기록한다. 연결이 다시 시도되어 서버용 ASDAS와 로컬용 ASDAS가 재 연결 되었을 때 현재 측정된 실시간 데이터가 전송되고 동시에 새로운 포트를 사용하여 서버와 접속하여 로컬용 ASDAS의 NAND 메모리에 기록하고 있던 덤프 데이터 또한 서버로 전송되어 날짜와 시간에 맞춰 파일로 저장한다. 그림 13은 재 연결을 위한 소스 코드를 나타내었다.

```
case SOCK_INIT: /* if a socket is initiated */
/* For TCP client's connection request delay : 3 sec */
if(time_return() - presentTime >= (tick_second * 3)) {
/* Try to connect to TCP server(Socket, DestIP, DestPort) */
printf("\r\n[ Attempt to connect on socket %d ]", mSocket);
connect(mSocket, Chconfig_Type_Def.destip, Chconfig_Type_Def.port);
presentTime = time_return();
}
break;
```

그림 13 SOCK\_INIT 상태에서 재 연결 시도  
Fig. 13 Retry connection in SOCK\_INIT state

로컬용 ASDAS에서 접속이 되지 않은 상태에서의 Socket 상태는 SOCK\_INIT이며 미리 정의해 놓은 카운터 값이 될 때마다 connect 함수를 실행시켜 서버와의 재접속을 시도한다. 로컬용 ASDAS가 서버용 ASDAS에 일단 접속이 되면 로컬(클라이언트) 또는 서버인지에 따라 각기 다른 플래그를 사용하여 데이터 전송준비를 한다. 여러 종류의 플래그는 지속적으로 확인되고 타이머에 의해 설정된 시간 간격 순으로 측정된 데이터를 정해진 포맷으로 전송한다. 데이터 전송 형식은 'YYYYMMDD\_HHMMSSMM\_XDATA\_YDATA\_ZDATA\_ATFCbit'와 같이 '날짜-시간-측별 데이터-이벤트검출유무'의 형식으로 전송된다. 이를 위해 링 버퍼(ring buffer)를 설계하여 sprintf 메시지를 통해 정해진 포맷으로 해당 데이터를 버퍼에 할당한다. 설계한 링 버퍼에서 데이터 전송형식이 완성되면 서버로 데이터를 전송하기 전에 위의 Socket 상태 확인 단계를 한 번 더 거친다. 이를 통해 네트워크상의 클라이언트는 서버에 데이터를 전송할 때 접속 상태를 한 번 더 확인하여 연결이 끊어졌을 때를 대비하게 된다.

서버용 ASDAS에서는 로컬용 ASDAS(클라이언트) 측과는 달리 다 수의 클라이언트가 접속할 경우를 기본적으로 가정하므로 소켓을 통해 네트워크상의 클라이언트를 구별하고 포트 번호로 다시 한 번 어떤 클라이언트가 접속했는지를 구별하게 된다. W5200 모듈은 자체 제원 상 8개의 소켓을 지원하고 16KB의 소켓버퍼를 유동적으로 각 소켓에 할당할 수 있다. 제안한 ASDAS 설계에서는 8개의 소켓에 2KB씩을 할당하여 최대 6개까지 클라이언트를 동시에 연결하여 사용할 수 있게 하였다.

### 3.3 지진 데이터 관리 방안

제안한 ASDAS에서 측정된 지진데이터를 저장하기 위해 범용 Micro SD 카드와 NAND Flash 메모리를 사용한 파일 시스템을 설계하였다. 각각의 저장매체는 정해진 기능을 수행하도록 구분하였다. 먼저, Micro SD 카드는 정해진 형식에 따른 측정 데이터 보관과 시스템 상시 로그 기록을 위해 사용되었다. 최상위 폴더에 'YYYYMMDD'와 같이 년월일 방식으로 폴더를 만들고 그 아래에는 한 시간 간격으로 폴

더를 24개를 생성한다. 각각의 폴더 아래에는 3분 단위의 로그 기록 파일이 저장되는 형식으로 총 20개의 파일이 생성된다. 이를 제안한 시스템에 적용하기 위해서 시스템 최초 부팅 시에 GPS 또는 사용자 프로그램으로부터 입력받은 현재 날짜와 시간 정보를 가지고 폴더와 파일을 생성한다. 해당하는 펌웨어 상의 루틴은 그림 14와 같다.

```
char* CreateDirectoryAccordingly(int year, int month, int day, int hour) {
int mYear, mMonth, mDay, mHour;
mYear = year; mMonth = month; mDay = day; mHour = hour;

printf("\r\nf_mkdir(0:/%04d%02d%02d) : ", mYear, mMonth, mDay);
sprintf(dirPath, "0:/%04d%02d%02d", mYear, mMonth, mDay);
res = f_mkdir(dirPath);
FPrintFatResult(res);

switch(mHour) {
case 0:
printf("\r\nf_mkdir(0:/%04d%02d%02d/00H-01H) : ", mYear, mMonth, mDay);
sprintf(dirPath, "0:/%04d%02d%02d/00H-01H", mYear, mMonth, mDay);
res = f_mkdir(dirPath);
FPrintFatResult(res);
break;
}
```

그림 14 FatFS 파일 시스템의 디렉토리 생성 루틴  
Fig. 14 Routine for the directory creation of FatFS file system

설계한 ASDAS에서는 파일 생성 시 이어쓰기가 가능한 모드(Append mode)를 사용하였다. ASDAS 설계에 사용한 MCU(STM32F103VCT6)는 256KB 플래시 메모리를 지원하지만 100Hz의 측정 데이터를 3분 동안 버퍼링(buffering)하면 데이터의 양이 내장 플래시 메모리의 용량을 초과하므로 한번에 Micro SD카드에 옮겨 저장 할 수 없다. 따라서 MCU에서 일정 시간에 해당하는 데이터를 버퍼에 보관하고 있다가 SD카드에 생성된 파일의 마지막 부분에 이어 쓰고 파일 전체 시간이 3분이 되면 파일을 닫아 새로운 파일을 생성하도록 하였다. 그림 15는 Append 모드 방식의 파일 쓰

```
void CreateFileAppendModeAccordingly(char *dirPath, int min) {
int mMin;
mMin = min / 3;

char filePath[31];
strcpy(filePath, dirPath);
printf("\r\nopen_append(&fsrc, filePath) : ");

switch(mMin) {
case 0:
strcat(filePath, "/EQLOG1.TXT");
// Create log file on the drive 0
res = open_append(&fsrc, filePath);
FPrintFatResult(res);
if(res == FR_OK) {
// Write buffer to file
//res = f_write(&fsrc, HEADER, sizeof(HEADER), &br);

printf("\r\nEQLOG1.txt successfully created");

// Close file
//f_close(&fsrc);
} else if (res == FR_EXIST) {
printf("EQLOG1.txt already exist");
}
break;
}
```

그림 15 Append 모드를 위한 소스코드의 일부  
Fig. 15 Some of source code for Append mode

기를 위한 소스코드의 일부를 보이고 있다.

그림 15에서 분 단위를 매개변수로 받아 매 3분이되면 open\_append 메서드를 통해 이어쓰기 모드로 파일을 생성한다. open\_append 메서드의 매개변수 중 'filePath'는 앞서 디렉토리(directory) 생성루틴에서 건네받은 디렉토리 문자열이다. 데이터 버퍼는 매초마다 비워지고 기록하는 동안 또 다른 데이터 버퍼가 실시간 데이터를 보관하면서 이어 쓰여질 준비를 하게 된다. 이는 MCU에서 데이터 버퍼 부담을 감소시키고 한 데이터 버퍼가 비워지는 동안 다른 데이터 버퍼가 측정된 데이터를 보관하여 데이터의 손실도 방지 할 수 있게 한다.

#### 4. 성능평가

제안한 방법의 유효성을 검증하기 위해 실제 지진신호와 동일한 진동을 만들어주는 San-Esu사의 가진기(SSV-125L)를 사용하여 시각 동기화 측면에서 설계·구현한 ASDAS의 데이터 취득 성능과 데이터 전송 및 저장 성능을 평가하였다. 그림 16은 고층 건물에서의 지진방재시스템 시나리오를 고려하여 구축한 통합 UEDPS의 구성을 보이고 있다. UEDPS는 상층용 ASDAS(로컬), 하층용 ASDAS(로컬), 건물 밖 자유장(대지에 위치)용 ASDAS(서버용 주 시스템), 그리고 원격 모니터링 시스템으로 구성되며 각각의 ASDAS와 모니터링 시스템은 네트워크 공유기(Hub)로 연결되어 있다.

그림 17은 그림 16의 UEDPS를 구성하는 ASDAS의 프로토타입 시스템과 모니터링 시스템용 사용자 프로그램이다. ASDAS는 그림 2의 시스템 블록도를 기반으로 구현되었으며 모니터링 시스템(Notebook PC)의 사용자 프로그램은 C#기반으로 작성되었다. 사용된 MEMS 가속도 센서는 14비트 디지털 출력을 갖는 Freescale사의 MMA8451Q이며 8개의 센서로 구성된 센서배열을 적용하였다. 시각 동기화를 위해 사용한 GPS 모듈은 (주)아센코리아의 AKS650이며 시스템의 정밀한 시각 관리를 위한 RTC의 외부 클럭은 32.768KHz의 크리스탈이 사용되었다. 또한 각 ASDAS에는 지진 발생 시 건물 붕괴 등으로 인한 전원공급 차단상황을 고려하여 리튬 배터리를 적용하여 최대 30분 ~ 1시간동안 비상전원이 공급되도록 설계되었다. 또한 리튬 배터리와 코인 배터리(CR2032)를 추가로 적용하여 ASDAS의 안정적인 시각 관리가 가능하도록 하였다. ASDAS에서 가속도 센서

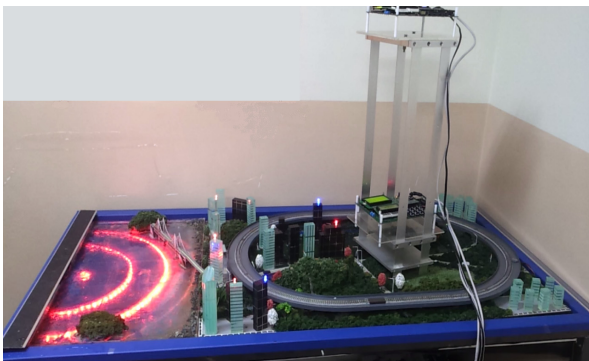


그림 16 고층 건물에서 UEDPS 성능 평가를 위한 테스트베드  
Fig. 16 Test bed for performance evaluation of UEDPS in high-rise building

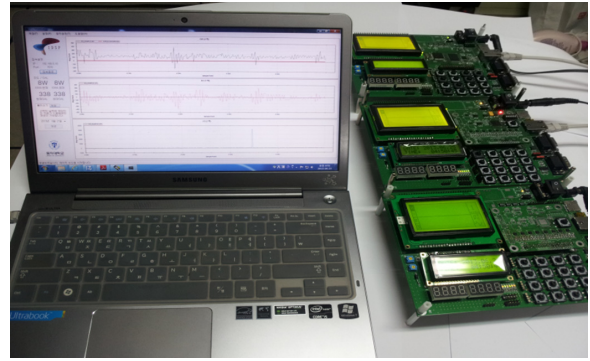


그림 17 UEDPS를 위한 ASDAS 및 모니터링 시스템  
Fig. 17 ASDAS and monitoring system for UEDPS

는 RTC 클럭을 사용하여 FPGA내 설계된 분주기를 통해 생성된 샘플링 주파수를 사용하여 지진 데이터를 취득한다. ASDAS의 데이터 취득 성능 평가에 사용된 입력신호는 가진기를 통해 생성한 10Hz 100gal과 15Hz 100gal의 정현파 신호이다. 이때 사용된 100gal의 가속도 값(정현파 진폭)은 센서별 시각 동기화 성능 평가 시 MEMS 센서의 고유한 부가잡음의 영향을 최소화하기 위해 선택한 값이다.

그림 18은 가진기를 사용하여 ASDAS에 10Hz 100gal 정현파 진동을 가한 후 각 MEMS 센서 출력을 위상이 0(zero)인 실제 사인파(theoretical sine wave)와 비교한 결과를 보이고 있다. 또한 표 1은 10Hz 및 15Hz 정현파 진동의 각 센서별 취득 오차율(여기서 오차율은 10초간 수집한 측정값과 이론값의 오차의 평균을 100gal로 나눈 값의 백분율로 계산함)을 나타낸 것이다. 각 MEMS 센서로부터 취득된 데이터는 잡음제거 등 다른 처리과정에 의한 시각 오차 영향을 받지 않도록 FPGA내 별도로 구현한 HDL기반의 데이터 취득 모듈을 통해 따로 수집하여 평가하였다. 그림 18과 표 1의 결과에서 발생한 오차는 MEMS 센서가 갖는 고유잡음 및 기계식 가진기가 갖는 진동 오차(vibration error)에 의한 것으로 추정되며 결과로부터 데이터 취득 시 RTC 클럭에 기반한 센서별 취득 시각 동기는 유효하다고 볼 수 있다.

그림 19는 두 개의 ASDAS를 가진기 상에 설치하고 동일한 진동에 대한 취득 데이터를 위상이 0(zero)인 실제 사인파와 비교한 결과이다. 각 ASDAS는 가진기 밖에 설치된 서버용 ASDAS에 연결되어 있으며 로컬용 ASDAS에서 취득된 데이터는 이더넷을 통해 서버용 ASDAS로 전송된다. 이때 서버용 ASDAS에서 수집한 데이터는 로컬용 ASDAS에서 미소중합처리 후 전송된 데이터이며 서버용 ASDAS의 위치중합처리를 수행하기 전 데이터이다. 그림 19의 결과로부터 제안한 데이터 전송 방법을 통해 취득된 데이터는 시각 동기 및 전송 오차가 발생하지 않았음을 확인할 수 있다.

그림 20과 그림 21은 UEDPS을 위해 구현한 모니터링 시스템용 사용자 프로그램과 파일이 저장된 결과화면이다. 설계한 모니터링 프로그램은 각각의 로컬용 ASDAS에서 취득한 지진 데이터 또는 서버용 ASDAS에서 수집한 지진 데이터를 원격지에서 실시간 확인할 수 있으며 파일 시스템을 통한 저장이 가능하도록 구현되었다. 또한 각 ASDAS의 주요 시스템 파라미터와 사용되는 알고리즘의 파라미터의 원격 설정이 가능하다.

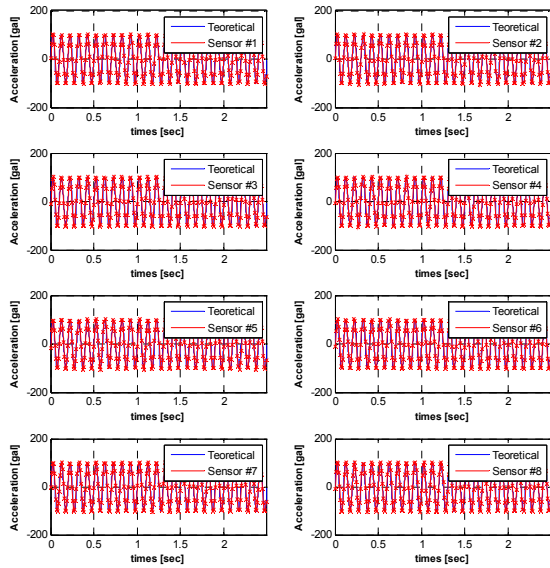


그림 18 ASDAS에서 센서별 취득데이터(10Hz 100gal)  
 Fig. 18 Measured data of each sensor in ASDAS(10Hz 100gal)

표 1 센서별 취득데이터의 오차율  
 Table 1 Error rate of seismic data measured by each sensor

센서 번호	취득 데이터의 오차율 [%/100gal]	
	10Hz	15Hz
1	4.69	5.02
2	4.78	4.28
3	4.92	4.27
4	4.96	5.27
5	5.43	5.31
6	4.76	4.73
7	5.35	6.10
8	5.61	6.81
STD	0.35	0.88

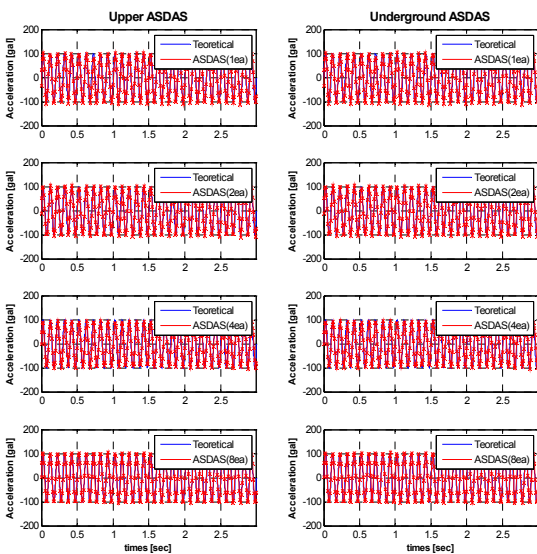


그림 19 서로 다른 ASDAS의 취득데이터(10Hz 100gal)  
 Fig. 19 Measured data from different ASDAS(10Hz 100gal)

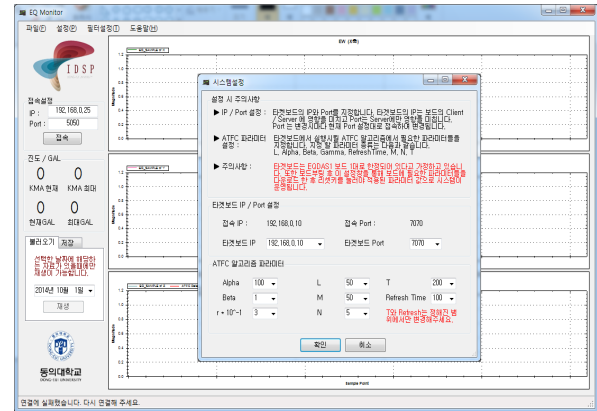


그림 20 UEDPS의 모니터링 시스템용 사용자 프로그램  
 Fig. 20 User program for monitoring system in UEDPS

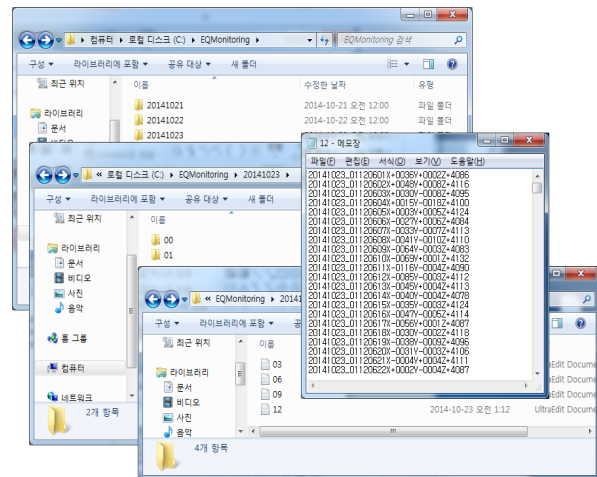


그림 21 저장 파일의 결과 화면  
 Fig. 21 Result screen of save file

### 5. 결론

본 논문은 도시형 지진방재시스템 (UEDPS; Urban Earthquake Disaster Preventing System)의 설계 및 구현 시 요구되는 GPS와의 시각 동기화와 정밀한 시각 관리 방안, 측정 데이터의 이더넷 전송을 위한 네트워크 구성 및 구현 방안, 그리고 수집된 데이터의 효율적 저장을 위한 방안을 제시하였다. 가진기를 이용하여 설계·구현한 UEDPS의 성능평가를 통해 제안한 데이터 취득, 전송, 그리고 저장 방법의 유효함을 검증하였다.

### 감사의 글

이 논문은 2013학년도 동의대학교 교내일반연구비 지원에 의해 연구되었음(2013AA139)

References

[1] J. Aschau and A. N. Kupperts, Early Warning Systems for Natural Disaster Reduction, Springer, 2003.

[2] Y. M. Wu and H. Kanamori, "Development of an earthquake early warning system using real-time strong motion signals," Sensors, vol. 8, pp. 1-9, 2008.

[3] R. M. Allen and H. Kanamori, "The potential for earthquake early warning in southern California," Science, vol. 300, pp. 786-789, May 2003.

[4] R. M. Allen, P. Gasparini, O. Kamigaichi and M. Bose, "The status of earthquake early warning around the world : an introductory overview," Seismological Research Letters, vol. 80, pp. 682-693, Sep. 2009.

[5] H. C. Ji. and et. al., Design of National Earthquake Early Warning System and Expanding of Realtime Data Sharing, Research rep., KIGAM, 2010.

[6] H. Choi and H. D. Bae, "Design and Implementation of Seismic Data Acquisition System using MEMS Accelerometer," Jour. of KIEE, vol. 61. no. 6, pp. 851-858, 2012.

[7] H. Choi, "Earthquake Event Auto Detection Algorithm using Accumulated Time-Frequency Changes and Variable Threshold," Jour. of KIEE, vol. 61. no. 8, pp. 1179-1185, Sep. 2012.

[8] H. Choi, S. U. Sohn, and H. D. Bae, "Seismic Noise Reduction Using Micro-Site Array Stacking," Jour. of KIEE, vol. 63. no. 3, pp. 395-403, Mar. 2014.

[9] STM Application note (AN3371) from [http://www.st.com/web/en/resource/technical/document /application\\_note /DM000 25071.pdf](http://www.st.com/web/en/resource/technical/document/application_note /DM000 25071.pdf)



윤 주 상(Joosang Youn)

2001년 고려대학교 전기전자전파공학부 공학사  
 2003년 고려대학교 전자공학과 공학석사  
 2008년 고려대학교 전자컴퓨터공학과 공학박사  
 2008년 3월 ~ 현재 동의대학교 멀티미디어 공학과 부교수  
 관심분야: 이동통신, 사물지능통신  
 Tel : 051-890-1993



허 경 용(Gyeongyong Heo)

1994년 연세대학교 전자공학과 공학사  
 1996년 연세대학교 전자공학과 공학석사  
 2009년 University of Florida 컴퓨터공학과 공학박사  
 2012년 3월 ~ 현재 동의대학교 전자공학과 조교수  
 관심분야: 인공지능, 패턴인식, 로봇공학  
 Tel : 051-890-1675

저 자 소 개



최 훈(Hun Choi)

1996년 충북대학교 전자공학과 공학사  
 2001년 충북대학교 전자공학과 공학석사  
 2006년 충북대학교 전자공학과 공학박사  
 1995년 ~ 1998년 LG반도체  
 2006년 ~ 2008년 한국표준과학연구원 Post-Doc.

2008년 3월 ~ 현재 동의대학교 전자공학과 부교수  
 관심분야: 적응신호처리, 계측신호처리  
 Tel : 051-890-1673