

고정밀 가속도 센서를 이용한 휴대용 진동 데이터 수집 시스템을 위한 통신 프로토콜 설계 및 개발

우경민*, 이현준**, 김수희*

*호서대 컴퓨터공학과, **RMS 테크놀로지

e-mail: wtos2k@gmail.com, lhj@rmstech.co.kr, shkim@hoseo.edu

Design and development of communication protocol for a portable vibration data collection system using high precision acceleration sensors

Kyungmin Woo*, Hyunjun Lee**, Suhee Kim*

*Dept of Computer Engineering, Hoseo University,

**RMS Technology Co., Ltd.

요 약

고정밀 가속도 센서를 이용한 휴대용 진동 데이터 수집 시스템은, 매우 고가이며 취급에 어려움이 존재하는 기존의 시스템과는 달리, 상대적으로 휴대가 용이하여 언제 어디서나 진동 데이터의 측정이 가능한 시스템으로, 향후 기능의 향상을 통해 기존의 시스템으로 측정이 불가능한 상황에서도 측정이 가능할 것으로 기대하는 시스템이다. 본 시스템은 데이터 수집기와 모바일 어플리케이션으로 구성되며, 이들 사이는 클래스1을 지원하는 고성능 블루투스 모듈을 통해 무선 통신한다.

본 논문에서는 개발한 진동 데이터 수집 시스템의 통신 시스템을 소개하고, 이 통신 시스템에서 사용하기 위한 프로토콜을 설계하고 개발한다. 4개의 채널에서 400Hz 속도로 발생하는 대량의 측정 데이터를 무선으로 전송하기 위해 패킷 구조를 사용하고, 데이터의 무결성을 위해 에러 검출 알고리즘을 사용한다. 이에 발생하는 문제해결을 위해 투과처리 및 역 투과처리 알고리즘을 사용한다.

1. 서 론

진동 데이터는 시간 도메인 형식으로 먼저 저장되고, 다음으로 주파수 도메인 형식으로 변환되어 분석된다. 전 세계적으로 진동 측정 산업이 빠른 속도로 확장되어 가는 추세에 따라, 진동 데이터의 분석을 위해 측정 데이터를 데이터베이스에 저장한 후 원격지에서 분석하는 시스템의 필요성이 증대되고 있으며, 구조적인 한계로 인해 측정 시스템의 구축이 곤란할 경우나, 고가의 DAS(Data Acquisition System)를 사용하지 않기 위한 대안으로 무선 측정 시스템의 필요성이 점점 증대되고 있다.

이러한 필요성에 따라 호서대학교와 RMS 테크놀로지는 고정밀 가속도 센서를 이용한 휴대용 진동 데이터 수집 시스템을 설계하고 개발하였다[1]. 개발한 시스템은, 기존의 컴퓨터와 유선으로 연결되는 센서-DAQ(Data Acquisition)-DSP(Digital Signal Processor)로 구성된 DAS와는 달리, GPS 모듈과 고성능 블루투스[2] 모듈이 탑재된 데이터 수집기를 이용하여 모바일 어플리케이션에 무선으로 데이터를 송신하여 분석하는 시스템이다. 개발한 데이터 수집기와 모바일 어플리케이션은 블루투스 모듈을

통해 무선으로 데이터를 송신하고 수신한다. 본 논문에서는 두 기기의 블루투스 통신을 위해 통신 프로토콜을 설계하고 구현한 내용에 대해 집중적으로 서술하고자 한다. 본 논문에서 소개하는 통신 프로토콜[3]은 이러한 무선 데이터 전송 시스템에서 대량으로 전송되는 데이터의 무결성을 위해 설계되었으며, 시리얼 통신을 시뮬레이션 하는 블루투스 프로토콜 스택 위에서 동작하는 어플리케이션 계층 단위의 프로토콜이다.

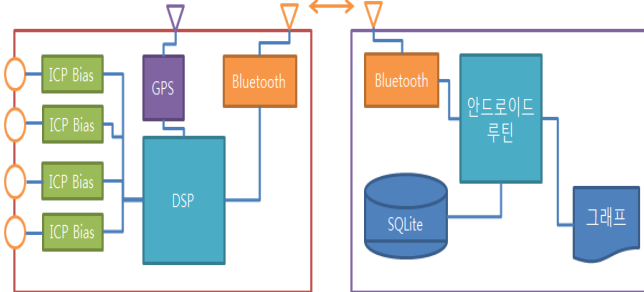
제 2장에서는 고정밀 가속도 센서를 이용한 휴대용 진동 데이터 수집 시스템을 소개하고, 제 3장에서는 이 시스템에서 사용한 통신 프로토콜의 설계 및 구현에 대해서 서술한다. 마지막으로 제 4장에서 결론을 맺는다.

2. 고정밀 가속도 센서를 이용한 휴대용 진동 데이터 수집 시스템

(그림 1)은 개발한 고정밀 가속도 센서를 이용한 휴대용 진동 데이터 수집 시스템의 구성도이다. 시스템은 크게 두 파트로 나누어진다. 한 파트는 고정밀 센서와 블루투스 모듈을 결합하여 만들어진 휴대용 데이터 수집기 RH-DAQ

이다. 두 번째 파트는 센서 모듈에서 송신하는 블루투스 데이터 신호를 스마트 디바이스가 수신하여 데이터 처리하고 그래프로 표현하는 모바일 응용 애플리케이션이다.

- ▶ 진동 센서 데이터 수집기 RH-DAQ
- ▶ 안드로이드기반 모바일 응용 애플리케이션



(그림 1) 가속도 센서를 이용한 휴대용 진동 데이터 수집 시스템 구성도

2.1 RH-DAQ

데이터 수집기 RH-DAQ는 4개의 채널을 동시에 지원한다. 각 채널마다 신호증폭 및 아날로그 신호의 디지털 변환을 위해 정밀 정전류 드라이버와 ADC 칩 LMP7312를 사용하였다. 송신 거리에 따른 신뢰성을 유지하기 위해 파워 클래스1을 지원하는 고성능 블루투스 모듈 Parani BCD110du를 사용하였으며, 동기화를 위한 GPS 모듈은 AKMU2P를 사용하였다. 이것은 NMEA-0183 포맷으로 로그를 제공하는 고정밀/고감도의 초소형 GPS 모듈이다. 이는 한 개 혹은 여러 개의 RH-DAQ장비의 시간 동기화를 위해 사용한다. 이들 장치의 제어를 위해 300MHz로 동작하는 고성능 MCU(Micro Controller Unit) TMS320F28346을 사용하였다.

RH-DAQ의 MCU는 아날로그 데이터 컨버터 (ADC)를 제어하여 채널당 초당 400번의 센서 데이터를 4 바이트 크기의 정수 형태로 획득한다. 모바일 디바이스가 블루투스 모듈을 통해 연결된 후에, 모바일 애플리케이션 측에서 센서정보(센서 타입, 센시티비티 값, 게인 값 등)가 수신되면, 센서 정보와 GPS 동기화 시간, 센서 데이터를 프로토콜의 투과처리 알고리즘을 이용 패킷화해 921600 Baud로 동작하는 블루투스 모듈을 통해 전송한다.

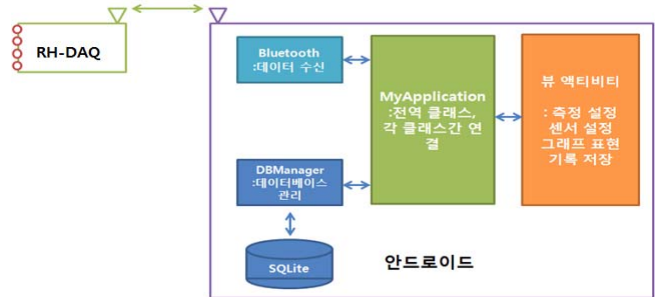
TMS320F28346의 프로세서는 명령어의 고속처리를 위해 데이터 메모리 어드레싱 회로가 단순하게 구현되어 있다. 그 결과, 바이트 단위 어드레싱은 불가능하며 워드 단위로만 어드레싱이 가능하다. TMS320F28346에서 사용 가능한 자료형의 최소 크기는 16비트로 고정되어 있으며. 이 후 소개할 프로토콜 구조는 이를 고려하여 설계되었다. (그림 2)는 TMS320F28346의 어드레싱 특성에 따른 자료형을 나타낸다.

Type	Hosted	C6000™	470	C55x™	C28x™
char	8	8	8	16	16
short	16	16	16	16	16
int	32	32	32	16	16
long	32	40	32	32	32
long long	64	-	-	40	-
float	32	32	32	32	32
double	64	64	64	32	32
long double	64	64	64	32	64

(그림 2) TMS320F28346의 자료형

2.2 안드로이드 기반 모바일 응용 애플리케이션

RH-DAQ 디바이스로부터 블루투스를 이용해 패킷을 수신한다. 프로토콜의 알고리즘 중 역 투과처리 알고리즘을 이용하여 원 데이터를 복원한다. 실시간 그래프 플로팅 알고리즘을 이용해 측정값을 화면에 출력하며, SQLite 데이터베이스에 데이터를 저장한다. 안드로이드 기반 모바일 응용 애플리케이션의 구성도는 다음과 같다.



(그림 3) 안드로이드 기반 모바일 응용 애플리케이션 구성도

3. 통신 프로토콜

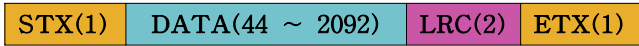
개발한 진동 데이터 수집 시스템은 RH-DAQ와 모바일 응용 애플리케이션으로 구성되어 있다. 두 모듈은 각각에 장착되어 있는 블루투스 모듈을 통하여 921600 Baud 속도로 무선통신 한다. 이 절에서는 개발한 RH-DAQ에 기반하여 블루투스 통신 프로토콜을 설계하고 구현한다. 개발한 프로토콜은 애플리케이션 계층 단위의 프로토콜이며, 무선 데이터 전송 시스템에서 대량으로 전송되는 데이터의 무결성을 위해 설계되었다.

3.1 프로토콜의 구조

프로토콜의 구조는 STX(Start Of Text), DATA, LRC(Longitudinal Redundancy Check), ETX(End Of Text)로 이루어져 있다. 패킷의 시작점을 알리기 위해 STX 문자가 패킷의 시작에 위치하며, 패킷의 끝을 알리기 위해 ETX 문자가 패킷의 마지막에 위치한다. 데이터의 에러 검출을 위해 ETX 앞에 LRC[4] 문자를 추가한다.

패킷의 크기는 수신부에서 패킷의 저장시 안전성과 프로세서의 어드레싱 특성을 고려하여 2의 배수로 고정하였다.

<표 1> 통신 프로토콜 구조



- STX [Start Of Text] : 패킷의 시작, 16진수 0x02 사용
- DATA : 헤더와 4개의 채널 데이터로 구성
- LRC [Longitudinal Redundancy Check] : 데이터의 에러 검출
- ETX [End Of Text] : 패킷의 끝, 16진수 0x03 사용

패킷의 DATA는 HEADER와 4개의 채널 WAVE_CH1, WAVE_CH2, WAVE_CH3, WAVE_CH4로 이루어져 있다. HEADER는 센서의 수, 샘플링 주기, 측정하고자 하는 센서의 센시티비티, 게인 값 등으로 구성된다.

<표 2> DATA (44 ~ 2092 Bytes) 구조

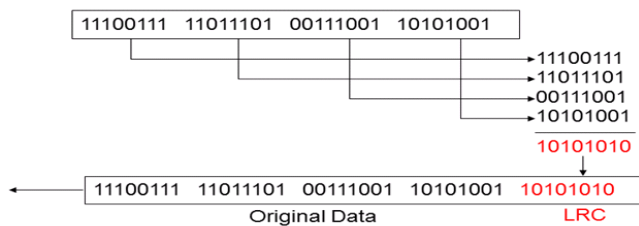
HEADER	WAVE_CH1	WAVE_CH2	WAVE_CH3	WAVE_CH4
(44)	(0~512)	(0~512)	(0~512)	(0~512)

<표 3> HEADER (44 Bytes) 구조

MsgId	# of Sensors	Sampling Freq.	최대 4개 Sensitivity 값	현재시간 (GPS)	데이터의 수	최대 4개 Gain 값
WORD (2)	WORD (2)	WORD (2)	FLOAT (4*4)	DWORD (4)	WORD (2)	FLOAT (4*4)

3.2 데이터의 에러 검출

데이터의 에러 검출을 위해, 에러검출 알고리즘 중, 데이터를 테이블화해 각 열에 대해 바이트 단위로 XOR 연산을 통해 패리티를 추가하는 방식인, 세로 중복 검사 방식(LRC)을 사용한다. RH-DAQ의 어드레싱 특성을 고려하고, 에러 검출 능력 향상을 위해 LRC의 크기를 2 바이트로 정하였다. (그림 4)는 바이트 단위의 데이터에 LRC 알고리즘을 적용한 예를 나타낸다.



(그림 4) LRC 알고리즘 적용의 예

3.3 투과처리 알고리즘

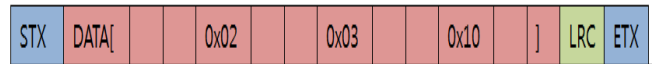
패킷의 STX와 ETX는 1 바이트로 표현이 가능한 16진수 0x00 ~ 0xFF중 0x02를 STX로, 0x03을 ETX로 사용하기로 약속한 것이다. 이 구조의 문제점은 패킷의 중간, 즉 데이터 부분이나 LRC 부분에도 0x02, 0x03이 존재할 수 있으며, 이 경우 수신 측에서 데이터의 0x02, 0x03을

STX, ETX로 간주할 수 있다는 것이다. 문제 해결을 위해 패킷이 전송되기 전 투과처리 과정을 거치며, 수신 후 역 투과처리 과정을 거친다.

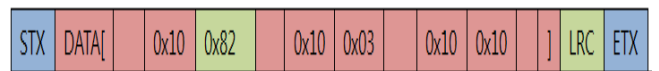
투과처리 알고리즘은 패킷의 STX, ETX 사이의 데이터 0x02, 0x03 앞에 전송 제어 확장 문자 DLE(Data Link Escape = 0x10)을 추가해 전송함으로써 이루어진다. 패킷의 데이터에 0x10이 존재할 경우를 대비해 데이터 0x10에도 투과처리가 이루어진다. 결론적으로 투과처리는 패킷의 STX, ETX 사이에 0x02, 0x03, 0x10에 대해 이루어진다.

수신부에서 패킷의 중간부터 수신 받는 경우, 투과처리된 데이터 0x10, 0x02 사이부터 수신될 때, 0x02를 STX로 간주하게 되는 문제점이 발생한다. 해결을 위해 투과처리 과정 중 0x02에 대해 MSB를 1로 변조해 패킷의 0x02가 STX에 유일하도록 만들어 해결한다. 이 후 수신 측에서 역 투과처리 과정 중 변조된 0x82를 0x02로 복원한다.

<표 4> 투과처리 전 패킷 구조

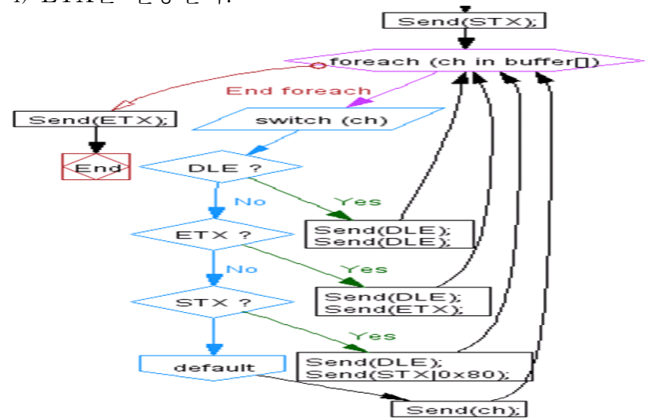


<표 5> 투과처리 후 패킷 구조



알고리즘에 대한 순서는 다음과 같다.

- 1) STX를 전송하고 2)로 간다.
- 2) 1바이트씩 데이터를 읽어와 전송하며 0x02, 0x03, 0x10 발견시 3)으로 간다. 데이터 전송이 끝나면 4)로 간다.
- 3) 0x03, 0x10시에는 DLE를 전송 후 전송하며, 0x02는 DLE전송 후 0x82를 전송한다. 2)로 간다.
- 4) ETX를 전송한다.



(그림 5) 투과처리 알고리즘 순서도

3.4 역 투과처리 알고리즘

역 투과처리 알고리즘은 투과처리 알고리즘을 이용하여 구성된다. 패킷의 STX가 유일하기 때문에 알고리즘의 시작은 STX를 찾는 것으로 시작된다. 패킷의 중간부터 수신되면, 사용할 수 없는 현재 패킷은 버리고 다음 패킷부터 처리를 시작한다. 알고리즘의 순서는 다음과 같다.

