DBpia

논문 16-41-03-09 The Journal of Korean Institute of Communications and Information Sciences '16-03 Vol.41 No.03
http://dx.doi.org/10.7840/kics.2016.41.3.338

단위 모듈 기반의 재조립 가능한 웨어러블 디바이스 구조 설계

이 거 윤◆, 강 순 주°

Design of Reassembly Unit Modular Wearable Device

Geo-Yun Lee◆, Soon-Ju Kang°

요 약

웨어러블 기기들은 사람이 착용하여 작동하고 서비스가 이루어지기 때문에 배터리 소모량, 크기, 무게 등 다양한 제약조건을 가지고 있다. 따라서 하나의 웨어러블 디바이스에 너무 많은 기능들을 부여하면 이러한 제약조건을 만족하지 못하여 실용성이 저하 될 뿐만 아니라 가격이 비싸져서 경쟁력을 상실하게 된다. 따라서 본 논문에서는 웨어러블 디바이스에 공통적으로 사용되는 여러 가지 기능을 재사용 가능한 모듈 형태의 독립된 디바이스로 재작하고, 사용자가 필요에 따라서 재조립하여 다양한 서비스를 받을 수 있는 시스템을 제안한다. 이 디바이스는 다수의 재조립 가능한 단위 모듈과 단위 모듈을 동적으로 장착할 수 있는 프레임으로 구성되어 있다. 프레임은 이 모듈간의 통신을 도우며 각각의 모듈들은 포함된 하드웨어에 따라 다양한 기능을 가지고 있다. 이를 구현하기 위해 본 논문에서는 각 서비스들이 필요한 모듈을 사용을 보장하고, 작업의 중요성에 따라 그 모듈 및 서비스에 우선순위를 부여하며, 사용하지 않는 모듈은 저전력 모드로 설정하여 단말의 저전력 동작이 가능하도록 설계했다.

Key Words : Modular Device, Unit Module-based, Wearable Device, Reassembly Device, Reassembly Module

ABSTRACT

Wearable Device has various constraint about battery power consumption, size, weight, etc, because the devices is worn and operated by person and provide services. So, if a device includes too many functions, it dose not satisfies the constraint and lose price competitiveness due to become expensive. Therefore we suggest that make reassembly Unit Modular Device witch has common used functions in wearable devices and user can receive various services to reassemble Unit Modules. It is comprised of frames and modules. Each module has various functions. Each frames help module to communicate each modules. To realize this device, we design to guarantee each services to use necessary modules, to give priority to modules depending on the important of the task, to set that does not use to low energy mode.

※ 이 논문은 2015년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임. [No. 10041145, 자율군집을 지원하는 웰빙형 정보기기 내장 소프트웨어 플랫폼 개발]
※ This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIP).[No. 10041145, Self-Organized Software platform(SoSp) for Welfare Devices]
◆ First Author : School of Electronics Engineering, College of IT Engineering, Kungbook National University, geoyun22k@gmail.com, 학생회원
° Corresponding Author : School of Electronics Engineering, College of IT Engineering, Kungbook National University, sjkang@ee.knu.ac.kr, 종신회원
논문번호 : KICS2015-11-353, Received November 1, 2015; Revised December 4, 2015; Accepted March 16, 2016

www.dbpia.co.kr

Ⅰ. 서 론

웨어러블 기기들은 다양한 센서와 통신 수단을 내장하여 사용자의 건강상태를 확인하고, 주변 IoT(Internet of Things) 디바이스들로부터 사용자 주변 정보를 받아 다양한 서비스를 제공할 수 있다. 예를 들어 웨어러블 기기에 내장된 가속도 센서를 이용하여 사용자의 움직임을 감지하고 [1] 제스처를 인식하여 디바이스의 제어를 하거나 사용자의 걸음걸이나 휴식 상태 등을 ECG(Electrocardiogram, 심전도)센서 혹은 SpO2(Pulse Oximeter Oxygen Saturation, 산소포화도)센서와 같은 각종 생체 센서[2]와 함께 분석하여 건강 상태를 확인 할 수 있다. 또한 무선 통신 장치를 이용하여 다른 웨어러블 디바이스들과 하나의 네트워크로 연결할 수 있다. 이것을 BAN(Body Area Network)라고 하며 주로 무선 네트워크를 이용하여 구성한다. 이 네트워크를 이용하여 각각의 웨어러블 디바이스가 가지고 있는 센서들을 이용하여 건강 모니터링 서비스나 응급상황 감지 서비스와 같은 서비스를 이용할 수 있다[3].

하지만 현재 웨어러블 디바이스는 착용 부위와 그 디자인에 따라 그 크기가 대부분 작기 때문에, 다양한 센서와 통신 장치들을 모두 내장 할 수 없다. 따라서 내장되어 있지 않은 장치를 사용하는 서비스들은 사용자에게 제공 할 수 없다는 문제가 있다[4]. 또한 무선 통신을 이용하여 네트워크를 구성할 때, 무선통신이 서로 간섭을 일으켜 통신 품질이 낮아지게 된다[4]. 그리고 다양한 뇌파 또는 심전도 신호와 같은 다양한 생체 신호가 동기화 되어 측정 되어야 한다. 하지만 무선통신을 이용하여 각 디바이스에서 신호들을 취득하게 되면 이 신호들의 동기화가 불가능하다.

본 논문은 사용자의 상황이나 요구에 맞게 재구성이 가능한 웨어러블 디바이스인 재조립 가능한 단위 모듈 기반 웨어러블 디바이스(이하 모듈 형 웨어러블 디바이스) 구조를 제안한다. 이 디바이스 구조는 웨어러블 디바이스가 가질 수 있는 기능들을 모두 모듈화하고, 그 모듈들을 사용자의 요구나 용도에 맞게 유선통신을 이용해 구성함으로서 맞춤형 웨어러블 디바이스를 구성할 수 있으며, 신체 각 부위에 자유롭게 부착하여 다수의 동기화된 생체 신호를 취득할 수 있다.

Ⅱ. 관련 연구

현재 웨어러블 디바이스는 그림 1과 같이 스마트 밴드, 스마트 시계, 스마트 안경 등과 같이 사용자의 특정 부위에 착용할 수 있는 형태의 디바이스이다. 이 웨어러블 디바이스들은 다양한 종류의 센서와 통신 모듈 등을 포함하고 있으며 포함된 하드웨어 종류에 따라 사용할 수 있는 서비스 종류도 달라질 수 있다.

웨어러블 디바이스 안에 들어갈 수 있는 하드웨어는 가속도 혹은 각속도 센서를 이용하여 사용자의 움직임을 인식하거나 예측할 수 있다.[5]

프로젝트 아라[6]는 이전 모토롤라에서 시작한 프로젝트이며 구글에서 진행 중인 오픈 하드웨어로 개발되는 모듈형 조립식 스마트폰 프로젝트이다. 이 프로젝트 아라의 스마트 폰은 그림 2에서 보듯이 크게 Module과 Endoskeleton으로 구성된다. 카메라, LTE 모뎀, AP, 저장장치, 디스플레이 등의 각각의 장치들을 모듈로 패키징하고, 그 모듈들을 Endoskeleton에 장착하여 하나의 스마트폰으로 동작할 수 있다.

하지만 이러한 재조립 가능한 모듈형 스마트폰의 경우 모듈 및 프레임의 구조상 두께의 경량화 제한 등으로 인해 구조적으로 스마트 폰의 경량화가 힘들다.

그림 1. 웨어러블 디바이스의 예 (스마트 시계, 스마트 안경)
Fig. 1. Examples of Wearable Device. (Smart Watch, Smart Glasses)

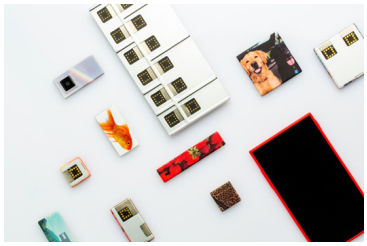

그림 2. Module과 Endoskeleton으로 구성된 프로젝트 아라의 사진.
Fig. 2. Figure of Project ARA consisting of Modules and Endoskeleton.

Ⅲ. 모듈형 웨어러블 디바이스 설계

3.1 시나리오 및 설계 요구 사항

그림 3은 본 논문에서 제안한 모듈형 웨어러블 디바이스를 이용하여 사용자의 용도나 기호 혹은 상황에 따라 사용할 수 있는 시나리오를 보여준다. 사용자의 착용 부위에 따라 모자, 허리띠, 밴드, 셔츠 등 다

www.dbpia.co.kr

그림 3. 모듈형 웨어러블 디바이스의 예시
Fig. 3. Examples of Modular Wearable Device.

양한 모양의 웨어러블 기기를 구성할 수 있다.

이 디바이스는 각종 모듈들과 본체를 이루는 프레임들로 구성되어 있다. 이 모듈들은 각각 독립적으로 동작하지 않고 프레임에 Plug in하여 동작할 수 있다. 그리고 여러 가지 모듈들을 장착한 웨어러블 디바이스는 구성된 모듈의 종류와 모양에 따라 웨어러블 디바이스 형태가 정해질 수 있다. 사용자의 운동 및 생활에 따른 Life Logging이 가능한 스마트 밴드, 사용자의 현재 위치에 따른 위치 기반 서비스를 제공 할 수 있는 스마트 벨트, 낙상감지 및 알림 서비스 및 실종 방지 서비스를 제공 가능한 스마트 모자, 환자의 상태를 주기적으로 확인하는 생체 모니터링 서비스가 가능한 스마트 셔츠 등이 될 수 있다.

본 논문에서 제안하는 단위 모듈 기반 웨어러블 디바이스 플랫폼은 다음과 같은 요구사항들을 만족해야 한다.

- 각 모듈들은 메인 모듈과 주변기기 모듈 역할이 가능하며 메인 모듈은 주변기기 모듈을 관리 및 제어할 수 있고, 주변기기 모듈로부터 데이터를 읽을 수 있다. 주변기기 모듈은 센서나 통신장치 등의 장치를 포함할 수 있으며 메인 모듈에 데이터를 제공할 수 있다.
- 메인 모듈과 주변기기 모듈 간의 Hot Plug 지원 가능한 통신 프로토콜이 필요하다.
- 모듈 간 통신을 위한 미들웨어와 어플리케이션 사이의 프로토콜이 필요하다.
- 멀티 태스크 환경에서 각종 데이터를 처리해야 하므로 태스크 수행 결정성 보장이 되어야 한다.
- 다양한 어플리케이션과 주변기기 모듈의 소프트웨어를 개발해야 하므로 이에 대한 서비스 개발의 용이해야 한다.
- 각종 생체신호(ECG, SpO2, 뇌파 등)들은 심박 혹은 뇌파 신호에 동기화 되어야 한다.
- 재사용 모듈을 무선으로 연동할 경우 신호 동기화 불가능 하므로 모듈 간 통신은 유선통신으로 해야 한다.

3.2 하드웨어 구조설계

본 논문에서 제안하는 단위 모듈 기반의 재조립 가능한 웨어러블 디바이스는 그림 4와 같이 다수의 프레임과 다수의 모듈로 구성되어 있다. 특히 모듈은 이 시스템 전체를 제어하고 관리하는 메인 모듈과 각 기능을 구성하는 주변기기 모듈로 구성되어 있다.

프레임은 다른 프레임과 연결되어 있다. 프레임은 모듈과 연결 할 수 있는 연결부(Connector)를 가져 모듈 간의 연결을 도와주고 전력을 공급하는 역할을 수행한다. 이 프레임은 연결부를 이용하여 각종 GPIO와 통신 버스를 연결 할 수 있다.

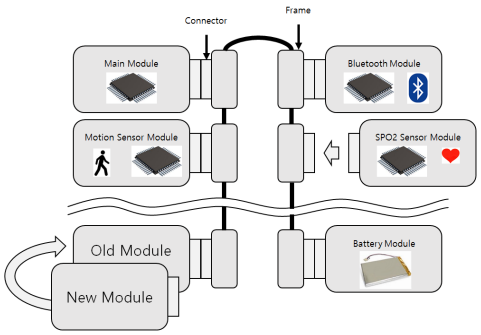

그림 4. 모듈형 웨어러블 기기의 하드웨어 구조도
Fig. 4. Hardware Architecture of Modular Wearable device.

3.2.1 모듈의 구조와 역할

각 모듈은 한 가지 이상의 장치를 가질 수 있으며 그 기능에 따라 역할이 달라진다.

그림 5와 같이 메인 모듈의 경우 이 디바이스의 주변기기 모듈을 관리 및 제어하는 역할이므로 Cortex M3 혹은 Cortex M4급 MCU가 필요하며, 충분한 IO가 필요하다. 또한 긴 배터리 수명을 필요로 하므로 저전력 제어가 가능해야 한다.

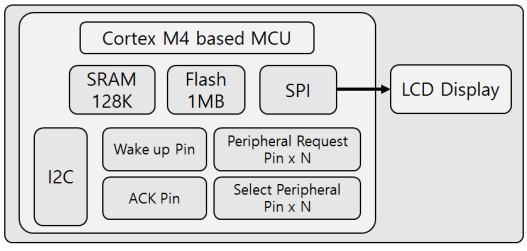

그림 5. 메인 모듈 하드웨어 구조의 예
Fig. 5. Example of Architecture of Main Module Hardware.

www.dbpia.co.kr

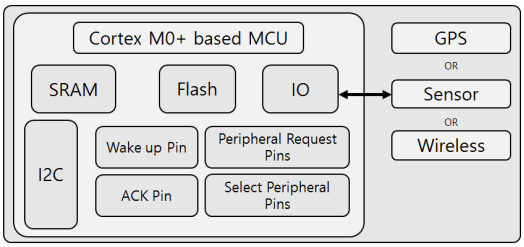

그림 6. 주변기기 모듈 하드웨어 구조의 예
Fig. 6. Example of Architecture of Peripheral Module Hardware.

그림 6은 MCU 이외의 구성에 따라 역할이 달라지는 주변기기 모듈의 구조의 예를 보여준다. 본 논문에서 보여주는 주변기기 모듈은 단위 기능별로 구분될 수 있다. 예를 들어 PPG센서 같은 생체 센서를 내장한 경우 이 모듈은 PPG 센서 모듈이 된다. 무선 통신 장치를 내장하는 경우 무선 통신 모듈이 된다. 그리고 GPS 장치를 내장하면 위치인지 모듈이 된다.

주변기기 모듈은 메인 모듈과 통신하고 관리되어져야 하므로 자체 MCU를 내장하고 있다. 그리고 저전력 기능을 포함한 CortexM0+ 동급 이상의 MCU가 필요하다.

3.2.2 프레임의 구조

프레임의 경우 모듈 간 연결을 도와주고 전력을 공급하는 역할을 수행한다. 그림 7과 같이 각 모듈의 통신을 위한 GPIO Pin들은 Wake Up Pin, Peripheral Request Pin, Select Peripheral Pin, ACK Pin 등이 있고, I2C와 같은 데이터 통신을 위한 버스도 있다.

- Wake Up Pin : 주변기기 모듈이 저전력 모드로 비활성화 된 메인 모듈을 활성화 시키는 용도이며 버스 형태이다.
- Peripheral Request Pin : 주변기기 모듈이 메인 모듈에 통신을 요청하기 위한 Pin이다. Wake Up Pin 전에 신호를 보낸다. 그 후 메인 모듈로부터 Select Peripheral Pin을 통해 신호가 올 때까지 Pending 한다. 활성화된 메인 모듈은 Pending된 Peripheral Request Pin 중 한 모듈을 골라 통신을 진행한다. 이 Pin은 메인 모듈과 주변기기 모듈이 1 대 1 개별로 연결 되어 있다.
- Select Peripheral Pin : 활성화 된 모듈이 접속된 다수의 주변기기 모듈 중 하나를 선택하여 통신하는 Pin이다. 통신하기 원하는 주변기기 모듈에 해당하는 Select Peripheral Pin에 신호를 보내어 ACK Pin으로부터 신호가 오면 그 때부터 통신을 할 수 있다.

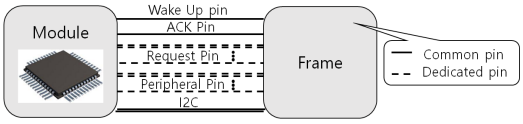

그림 7. 모듈과 연결된 프레임의 구조
Fig. 7. Structure of Frame connected to Module.

3.3 소프트웨어 구조 설계

앞의 시스템 설계 요구사항에서 언급했듯이 모듈 간 통신을 지원하는 미들웨어가 필요하고, 그 미들웨어와 Application 간의 프로토콜이 필요하다.

그림 8은 모듈형 웨어러블 디바이스에서 메인 모듈의 소프트웨어 구조이다. 이 구조는 어플리케이션 층(Application Layer), 시스템 층(System Layer), OS 층(OS Layer)으로 구성되어 있다.

메인 모듈의 어플리케이션 층은 사용자에게 서비스를 제공하기 위한 어플리케이션들을 포함하고 있다. 이 어플리케이션들이 주변기기 모듈과 통신하기 위해 모듈 간 통신 프로토콜에 등록해야 한다.

시스템 층에는 모듈 간 통신 프로토콜이 포함되어 있다. 이 프로토콜은 다수의 주변기기 모듈과 메인 모듈 간의 통신을 지원하는 프로토콜이다. 메인 모듈의 경우 모듈 관리자(Module Manager)와 전송 관리자(Transfer Manager)로 구성되어 있다. 모듈 관리자는 각 주변기기 모듈과 그 모듈과 관련된 어플리케이션을 관리하는 역할이다. 예를 들어 메인 모듈의 동작 중 실시간으로 접속 및 접속 해제를 관리하며 주변기기 모듈과 메인 모듈의 어플리케이션을 관리한다. 전송 관리자는 각 모듈과 통신을 하기 위한 역할이다.

메인 모듈의 경우 다수의 태스크가 동작하며, 환경

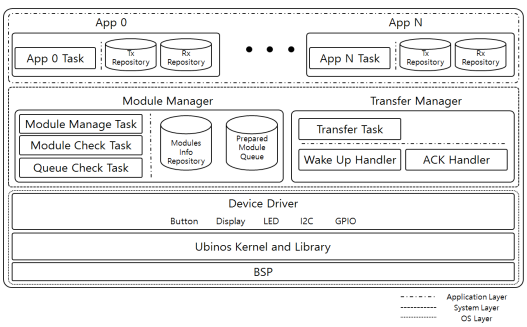

그림 8. 메인 모듈의 소프트웨어 구조
Fig. 8. Software Architecture of Main Module.

www.dbpia.co.kr

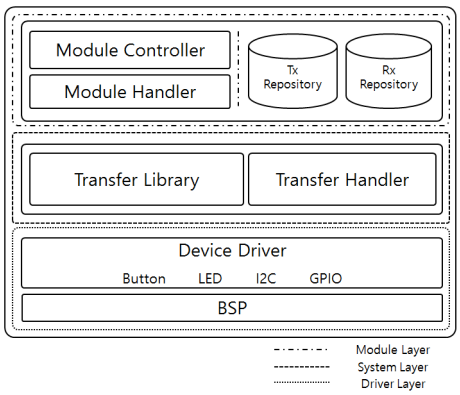

그림 9. 주변기기 모듈의 소프트웨어 구조
Fig. 9. Software Architecture of Peripheral Module.

변화 및 제어 등에 대해 실시간 성을 가져야 한다. 또한 웨어러블의 경우 자원 제약이 크고 적은 배터리를 가지는 시스템이므로 그에 맞는 실시간 운영체제를 가진다.

그림 9는 주변기기 모듈의 소프트웨어 구조이다. 모듈 층(Module Layer), 시스템 층(System Layer), 드라이버 층(Driver Layer)로 나뉜다. 모듈 층은 주변기기 모듈이 포함하는 센서나 장치 등을 제어할 수 있는 모듈 컨트롤러나 핸들러가 있다. 시스템 층은 메인 모듈과 주변기기 모듈간의 통신을 위한 모듈간 통신 프로토콜이 있다. 또한 이 프로토콜은 주변기기 모듈이 동작 중 메인 모듈과 Plug in을 지원한다. 드라이버 층은 각종 통신을 지원하는 드라이버를 내장하고 있다. 이 주변기기 모듈은 다수의 태스크를 필요로 하지 않으므로 OS 없이 동작한다.

3.3.1 모듈 간 통신 프로토콜 설계

모듈 간 통신 프로토콜은 다수의 주변기기 모듈과 메인 모듈의 통신을 지원하고 동작 중 주변기기 모듈의 Hot Plug를 지원하는 미들웨어이다. 또한 주변기기 모듈의 전원관리를 제어하여 저전력을 제어한다.

(1) 메인모듈이 주변기기 모듈과 통신하는 과정

그림 10은 메인 모듈이 주변기기 모듈과 통신하는 과정을 순서도로 표현한 것이다. 메인 모듈의 어플리케이션은 특정 주변기기 모듈과 통신하기 위해 모듈 관리자(Module Manage Task)에 요청을 보낸다①. 모듈 관리자는 관련된 주변기기 모듈이 메인 모듈에 접속이 되어 있는지 확인하고 통신 대기 큐에 저장한다②. 이 때 주기적으로 동작하는 큐 확인 태스크에서 모듈 대기 큐에 저장된 통신 정보를 확인 한 후③ 통

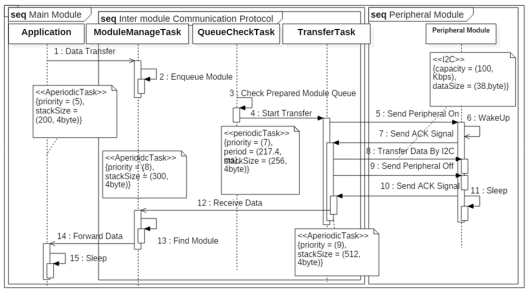

그림 10. 메인 모듈이 주변기기 모듈과 통신하는 과정을 나타낸 순서도
Fig. 10. Sequence Diagram of Communication Process that Main Module Communicates with Peripheral Module.

신 시작 요청을 보낸다④.

전송 관리자는 요청을 읽고 특정 주변기기 모듈의 ID와 해당 주변기기 모듈의 프레임 위치를 읽고 통신을 진행한다. 통신 과정은 다음과 같다.

요청에 포함된 주변기기 모듈의 ID와 프레임 위치에 따라 해당되는 Select Peripheral Pin에 Peripheral On 신호를 보내⑤ 저전력모드에 있던 주변기기 모듈을 깨우고⑥ ACK 신호를 기다린다. ACK 신호를 받은 후⑦ I2C를 이용하여 데이터를 송수신한다⑧. 데이터 통신이 완료 되면 해당 주변기기 모듈을 비활성화 하기 위해 Select Peripheral Pin에 Peripheral Off 신호를 보낸다⑨. 신호를 받은 주변기기 모듈은 ACK 신호를 메인 모듈에 보내고⑩, 저전력 모드로 바꾼다⑪. ACK신호를 받은 메인 모듈의 전송 관리자는 해당하는 주변기기 모듈과의 통신을 종료 한 후 모듈 관리자에 그 결과를 전송한다⑫.

모듈 관리자는 받은 통신 결과를 주변기기 모듈의 ID와 관련된 어플리케이션에게 전달한다⑬~⑭. 모든 작업을 완료한 메인 모듈은 저전력 모드로 전환한다⑮.

(2) 주변기기 모듈이 메인 모듈에 통신을 요청하여 통신하는 과정

그림 11은 주변기기 모듈이 메인모듈을 활성화 시킨 후 통신을 요청하는 과정이다. 주변기기 모듈은 메인 모듈과 통신하기 전 Peripheral Request Pin을 Pending① 시킨 후 Wake Up 신호를 이용②하여 메인 모듈을 저전력 모드에서 활성화시킨다③. 그 후 메인 모듈은 Pending된 Peripheral Request Pin을 읽고④ 현재 접속 중인 주변기기 모듈인지 확인 한다. 그리고 통신 우선순위에 따라 통신 정보를 대기 큐에 저장한다⑤. 통신 대기 큐에 저장된 우선순위와 순서

www.dbpia.co.kr

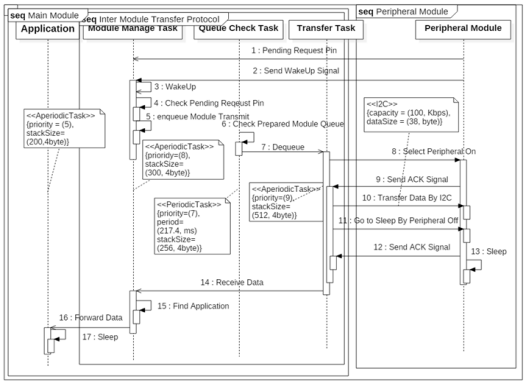

그림 11 주변기기 모듈이 메인 모듈에 통신을 요청하여 통신하는 과정을 나타낸 순서도
Fig. 11. Sequence Diagram of Communication Process when Peripheral Module request communication to Main Module.

에 따라 통신을 진행한다⑥ ~ ⑦. 해당하는 Select Peripheral Pin에 Peripheral On Signal을 보내고⑧ ACK를 받은 후⑨ 데이터 통신을 한다. 통신이 완료되면⑩ Select Peripheral Pin에 Peripheral Off Signal을 보내고⑪ ACK를 받으면⑫ 주변기기 모듈과의 통신을 종료한다. 그 결과를 모듈 관리자를 통해 해당하는 어플리케이션에 보낸다⑬~⑮. 모든 작업이 완료되면 메인 모듈은 저전력 모드로 바뀐다⑯.

(3) 동작 중 주변기기 모듈의 Plug in 과정

그림 12는 주변기기 모듈의 Plug in 과정을 나타낸다. 이 과정을 이용하여 메인 모듈이 동작 중 주변기기 모듈의 Plug in을 인식할 수 있다. 이 과정은 주변기기 모듈이 메인 모듈에 통신을 요청하여 통신하는 과정과 유사하다.

물리적으로 메인 모듈과 연결된 주변기기 모듈은 전원으로 인한 Reset 동작을 하고①, 그 즉시 주변기기 모듈이 메인 모듈에 통신을 요청한다② ~ ③. 메인 모듈이 Wake Up 되고④, Peripheral Request 신호를 읽는다⑤. 이 때 새로운 주변기기 모듈일 경우 메인 모듈의 모듈 관리자는 주변기기 모듈에 모듈 ID를 요구하는 통신 정보를 통신 대기 큐에 저장한다⑥. I2C 통신을 이용하여 주변기기 모듈의 모듈ID를 읽는다⑧ ~ ⑬. 주변기기 모듈은 저전력 모드로 바뀌고⑭, 관련 모듈 정보를 업데이트 한다⑮. 그 후 연결된 주변기기 모듈과 관련된 어플리케이션에 Plug in 정보를 준다⑯ ~ ⑰. 완료 후 메인 모듈은 저전력 모드로 돌아간다⑱.

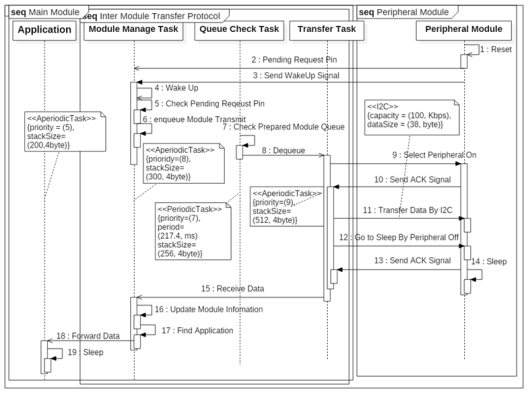

그림 12. 주변기기 모듈의 Plug in 과정을 나타낸 순서도
Fig. 12. Sequence Diagram of Plug-in Process of Peripheral Module.

(4) 동작 중 주변기기 모듈의 Plug out을 인지 하는 과정

그림 13에 대한 과정은 메인 모듈에 접속된 주변기기 모듈이 Plug out이 되었는지 주기적으로 확인하는 과정이다. 적어도 하나 이상의 주변기기 모듈이 Plug in 되었을 때 관련 Task가 동작한다①. 그 후 Select Peripheral Pin 신호와 그에 대한 ACK 신호를 이용하여 해당 주변기기 모듈이 메인 모듈에 접속 중인지 확인한다② ~ ③. 만약 Time Out이 될 때까지 ACK 신호를 받지 못한다면④ 해당 주변기기 모듈이 Plug Out이 된 것으로 확인하고 관련 주변기기 모듈의 정보를 업데이트 한다⑤. 그 후 관련된 어플리케이션에 해당 주변기기 모듈이 Plug Out 되었음을 알린다.

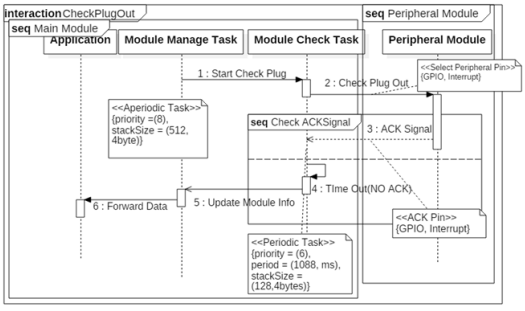

그림 13. 주변기기 모듈이 Plug out하는 과정을 나타낸 순서도
Fig. 13. Sequece Diagram of Plug-out Process of Peripheral Module.

www.dbpia.co.kr

Ⅳ. 구현 및 성능 평가

4.1 모듈형 웨어러블 디바이스 구현

그림 14는 앞에 설계한 모듈형 웨어러블 디바이스를 허리띠에 착용할 수 있는 스마트 벨트로 구현하였다. 가운데 메인 모듈이 자리 잡고 있으며, 주변기기 모듈은 양 옆으로 탈부착 가능하며 동작 중 Hot Plug가 가능하다.

그림 15는 스마트 벨트를 구성하는 모듈인 메인 모듈과 주변기기 모듈의 모습과 하드웨어 구조를 보여준다. 그림 15의 위의 그림은 메인 모듈이다. 메인 모듈은 Cortex M4와 RTOS인 UBINOS를 사용하며 주변기기 모듈들을 제어 및 관리하며 Hot Plug를 지원한다. 그림 15의 아래 그림은 주변기기 모듈이다. 주변기기 모듈은 CortexM0+를 사용하며 OS가 없다.

본 논문에서는 이 구조를 이용하여 사용자의 요구에 따라 사용자의 움직임을 측정하거나 사용자의 생체 정보를 얻을 수 있는 스마트 벨트를 구현하였다.

그림 16은 스마트 벨트에 모션 센서 기능을 가진 주변기기 모듈을 Plug In하여 사용자의 가속도 값을 측정하는 모습이다.

그림 17는 모션 센서 모듈을 장착한 스마트 벨트 사용하는 도중 SpO2 센서 모듈을 추가로 Plug in 한

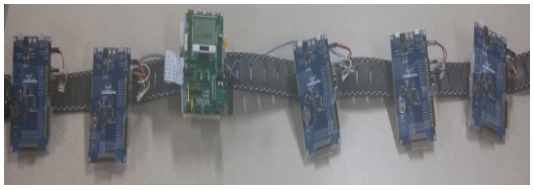

그림 14. 모듈형 웨어러블 디바이스로 구현한 스마트 벨트
Fig. 14. Smart belt implemented in a modular wearable devices.

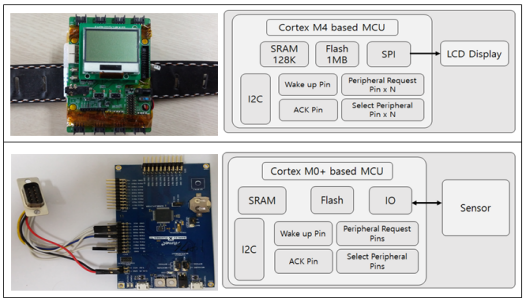

그림 15. 스마트 벨트의 메인 모듈 및 주변기기 모듈의 하드웨어 구조
Fig. 15. Hardware Structure of Main Module and Peripheral Module in Smart Belt.

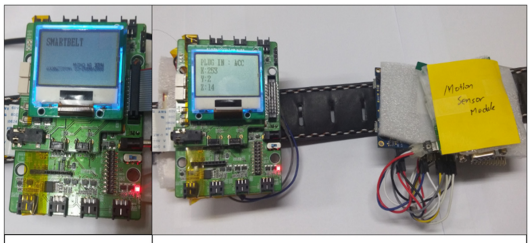

그림 16. 모션 센서 모듈을 스마트 벨트에 Plug In 했을 때 메인 모듈과 주변기기 모듈의 모습
Fig. 16. View of the Main Module and the Peripheral Modules when plug-In Motion Sensor Module in the Smart Belt.

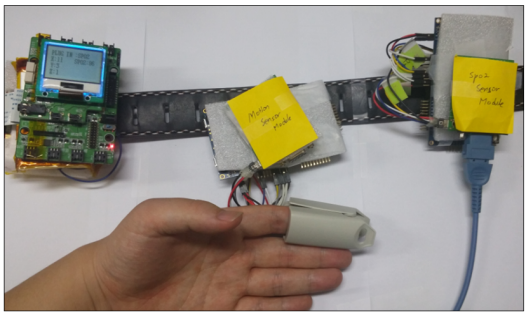

그림 17. 스마트 벨트 동작 중 SpO2 센서 모듈을 스마트 벨트에 Plug In 했을 때 메인 모듈과 주변기기 모듈들의 모습
Fig. 17. View of the Main Module and the Peripheral Modules of the Smart Belt when SpO2 Sensor Modules have Plug-in to the Smart Belt.

모습이다. 이 때 기존의 모션 센서 모듈로부터 가속도 값을 읽어 들이는 도중 Plug in이 가능하며 동시에 SpO2값을 읽는 것을 확인 할 수 있다.

그림 18은 필요가 없어진 SpO2 센서 모듈을 스마트 벨트에서 Plug Out을 이용하여 제거한 모습이다. 기존의 모션 센서 모듈은 계속 값을 읽는다.

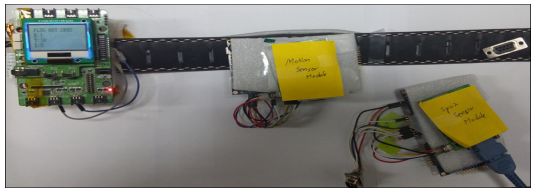

그림 18. 스마트 벨트 동작 중 SpO2 센서 모듈을 Plug Out 했을 때 모습
Fig. 18. View of the Main Module and the Peripheral Modules of the Smart Belt when SpO2 Sensor Modules have Plug-out to the Smart Belt.

www.dbpia.co.kr

4.2 스마트 벨트의 성능 평가

구현한 스마트 벨트의 메인 모듈이 다수의 주변기기 모듈을 관리하고 통신하는 성능을 평가한다. 1개의 메인 모듈과 총 5개의 주변기기 모듈을 이용하여 성능을 평가하며 각각 50회 실시하여 평균 소요 시간을 구한다. 각 주변기기 모듈에 대응하는 어플리케이션은 모두 같은 우선순위를 가지며 200ms의 주기를 가진다. 또한 실험에 진행되는 하나의 주변기기 이외의 주변기기 모듈들은 주기적으로 메인 모듈과 데이터 통신을 진행한다. 평가 과정은 다음과 같다.

4.2.1 모듈간 데이터 통신 전 모듈을 제어하는 과정

이 과정은 주변기기 모듈이 메인 모듈과 통신을 요청하기 위한 과정이 있으며 이는 Plug in과정이 포함되어 있다. 그리고 메인 모듈의 어플리케이션이 주변기기 모듈 중 하나를 선택하여 통신을 준비하는 과정이 있다.

4.2.2 모듈 간 데이터 통신 중, 통신 후 및 주변기기 모듈을 분리하는 과정

이 과정은 메인 모듈과 주변기기 모듈이 데이터 통신을 진행하는 과정과 그 결과를 해당하는 메인 모듈에 전달하는 과정이 있다. 그리고 주변기기 모듈이 Plug Out을 인식하는 과정이 있다.

4.3 실험 결과

4.3.1 모듈간 데이터 통신 전 모듈을 제어하는 과정

주변기기 모듈이 메인 모듈과 통신 요청하는 과정(Peripheral Request)은 모듈의 수와 관계없이 약 34ms ~ 37ms의 시간이 소요되었다.

메인 모듈의 어플리케이션이 주변기기 모듈과 통신 요청하는 과정(Select Peripheral)또한 모듈의 수와 관계없이 약1ms 내로 완료되었다. 따라서 메인 모듈과 주변기기 모듈의 통신 전 준비과정은 모듈의 수와 무관함을 알 수 있다.

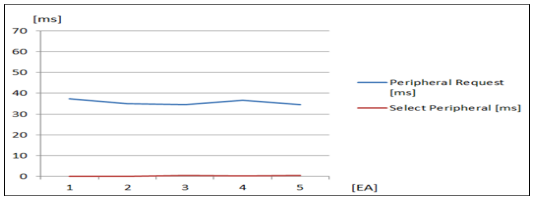

그림 19. 주변기기 모듈이 메인 모듈에 통신하기 까지 소요 시간을 그린 그래프
Fig. 19. Graph of Peripheral Module 's Communication completion time.

표 1. 주변기기 모듈이 메인 모듈에 통신하기 까지 소요시간
Table 1. Peripheral Module 's Communication completion time.

Number of Modules [EA]	1	2	3	4	5
Peripheral Request [ms]	37.4	35	34.5	36.7	34.5
Select Peripheral [ms]	0.12	0.12	0.512	0.383	0.579

4.3.2 모듈 간 데이터 통신 중, 통신 후 및 주변기기 모듈을 분리하는 과정

메인 모듈이 주변기기 모듈을 선택 후 데이터 통신하는 과정(Transfer Process)은 모듈의 수가 증가할 때 약 194ms 지수적으로 증가하여 주변기기 모듈의 수가 5개가 되었을 때 약 420ms가 소요됨을 알 수 있다.

모듈 간 통신 후 결과 데이터를 어플리케이션에 전달할 때 소요되는 시간은 모듈의 수가 증가할 때 소요시간도 약 4ms에서 약 420ms까지 증가함을 알 수 있다. 이는 통신에 참가하는 주변기기 모듈에 관련된 어플리케이션의 태스크의 수와 연관된다.

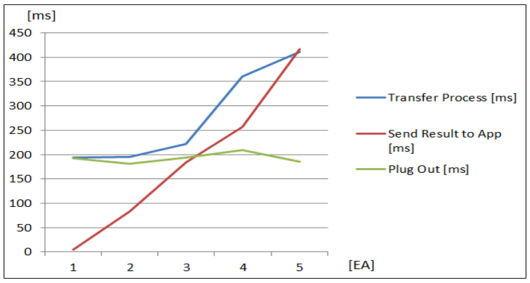

그림 20. 주변기기 모듈이 메인 모듈에 통신 중 및 통신 후 처리 소요 시간, 그리고 Plug Out과정을 완료 소요 시간을 그린 그래프
Fig. 20. Graph of Communication Time between Peripheral Modules and Main Module and Process Time of Plug-out.

표 2. 주변기기 모듈이 메인 모듈에 통신 중 및 통신 후 처리 소요 시간, 그리고 Plug Out과정을 완료 소요 시간
Table 1. Communication Time between Peripheral Modules and Main Module and Process Time of Plug-out.

Number of Modules [EA]	1	2	3	4	5
Transfer Process [ms]	193.6	195.2	222.4	360	411.6
Send Result to App [ms]	4	83.4	184.4	257.2	416.8
Plug Out [ms]	192	180.8	193.6	209.6	185.2

V. 결 론

기존 웨어러블 디바이스들은 크기가 제한되기 때문에 하드웨어 구성도 제한된다. 따라서 그 서비스를 제공할 수 있는 하드웨어가 존재하지 않으면 결국 착용자는 원하는 서비스를 받을 수 없게 된다. 또한 무선

www.dbpia.co.kr

으로 네트워크를 구성할 경우 서로 간섭을 일으켜 무선통신의 품질이 낮아지고, 다양한 센서 데이터들을 동기화시키기 어렵다. 그러나 본 논문에서 제시하는 웨어러블 디바이스는 착용자의 부착 위치나 그 용도가 변화함에 따라 교체함으로서 사용자가 원하는 서비스를 제공할 수 있으며, 유선통신을 이용하여 다수의 무선통신으로 인한 간섭 요인을 제거하고 각각의 센서 데이터를 동기화하기 쉽도록 설계하였다.

단위 모듈 기반의 재조립 가능한 웨어러블 디바이스는 각각의 주변기기 모듈을 관리, 제어 및 통신을 위한 모듈 간 통신 프로토콜을 제안하였다. 또한 이 프로토콜은 주변기기 모듈과 통신을 하기 위한 어플리케이션이 통신을 할 수 있도록 지원한다. 그리고 다양한 서비스를 지원하기 위한 어플리케이션을 개발할 수 있도록 계층화된 구조를 제안하였다.

제안한 재조립 가능한 웨어러블 디바이스를 검증하기 위해 SpO2 센서 모듈, 가속도 센서 모듈, ECG 센서 모듈 등 주변기기 모듈과 메인 모듈을 이용하여 모듈간 통신 프로토콜을 검증하였다. 그 결과 Peripheral Request 및 Select Peripheral 그리고 Plug Out 과정은 주변기기 모듈의 수와 관계없이 완료 시간이 일정하며 각각 약 40ms, 약 1ms, 약 210ms 내에 완료되는 것을 확인하였다. 반면 주변기기 모듈과 메인 모듈 간 데이터 통신 및 통신 결과를 어플리케이션에 전달하는 과정은 모듈의 수가 증가함에 따라 완료시간이 증가함을 알 수 있었다.

향후 연구 과제는 ECG나 SpO2와 같은 생체 센서 모듈과 통신 모듈 등과 같이 다양한 동작을 수행하는 주변기기 모듈을 구현한다. 또한 주변기기 모듈의 조합과 그와 관련된 다양한 어플리케이션 환경에서 중요성 따른 다양한 서비스를 설계하고 그 동작을 보장하는 것을 연구한다.

References

[1] P. Casale, O. Pujol, and P. Radeva, "Human activity recognition from accelerometer data using a wearable device," *Pattern Recognition and Image Analysis*, pp. 289-296, 2011.

[2] T. Klingeberg and M. Schilling, "Mobile wearable device for long term monitoring of vital signs," *Computer Methods and Programs in Biomedicine*, vol. 106 no. 2, pp. 89-96, May 2012.

[3] J. Ko, S. G. Hong, B. B. Lee, and N. S. Kim, "Trends of converging smart devices with IoT technology," *Electron. Telecommun. Trends*, vol. 28, no. 4, pp. 79-85, Aug. 2013.

[4] M. Chen, S. Gonzalez, A. Vailakos, H. Cao, and V. C. M. Leung, "Body area networks: A survey," *Mob. Netw. Appl.*, vol. 16, no. 2, pp. 171-193, Apr. 2011.

[5] C. Y. Yun, "A study on implementation of U-healthcare monitoring system in WBAN," *J. KICS*, vol. 35, no 8, pp. 171-178, Aug. 2010.

[6] Alex Ray, History of Project Ara and modular phones, Market pilot in Puerto Rico, Project Ara Developers Conference(2015), Retrieved April., 20, 2015, from http://www.modularphonesforum.com/news/history-of-project-ara-and-modular-phones-market-pilot-in-puerto-rico-359/

이 거 윤 (Geo-Yun Lee)

2014년 2월 : 경북대학교 IT대학 전자공학부 공학사

2016년 2월 : 경북대학교 IT대학 전자공학부 공학석사

<관심 분야> 실시간 운영체제, 실시간 시스템, 임베디드 시스템

강 순 주 (Soon-Ju Kang)

1983년 2월 : 경북대학교 전자공학과 공학사

1985년 2월 : 한국과학기술원 전자계산학과 공학석사

1995년 2월 : 한국과학기술원 전자계산학과 공학박사

1985년~1996년 : 한국원자력연구소 연구원, 핵인공지능연구실 선임연구원, 전산정보실 실장

2000년~2001년 : University of Pennsylvania, Dept. of Computer and Information Science, 객원 연구 교수

1996년~현재 : 경북대학교 IT대학 전자공학부 정보통신공학전공 정교수

2011년~현재 : 자율군집 소프트웨어 플랫폼 연구센터 센터장

<관심분야> 실시간 시스템, 소프트웨어 공학, 지식기반 시스템

www.dbpia.co.kr