

센서 중계 노드(SRN) 설계 및 구현

홍성웅*, 김윤서*, 김평중**

*(주)이알과워

**충북도립대학교 컴퓨터정보과

e-mail:leoking@cbpc.ac.kr

Design & Implementation of Sensor relay node

Sung-Woong Hong*, Youn-Seo Kim**, Pyeong-Jung Kim**

*Dept of Computer & Information Engineering, Cheongju University

**Dept of Computer Information, Chungbuk Provincial College

요 약

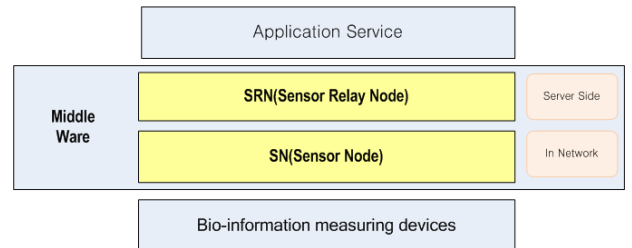
본 논문에서는 사용자자동인식, 실시간 상태 모니터링에 적합한 유헬스 시스템을 위한 생체정보측정기와 연계된 임베디드 미들웨어 SRN(Sensor Relay Node)를 제안한다. SRN은 U-health 서비스를 제공하기 위해 센서정보와 정보처리 기능을 갖으며 센싱 정보수집기와 어플리케이션에 처리된 정보를 제공한다.

1. 서론

인구의 급속한 고령화와 더불어 국민의 복지수준이 향상됨에 따라 건강문제에 대한 효율적인 해결방안으로 병원을 방문하지 않고 의료 서비스를 받는 U-Healthcare의 필요성이 증대되고 있다.^[1] 유헬스케어에서 신뢰성 있는 시스템^[2], 병원에서의 Contextaware mobile system^[3] 그리고 유헬스케어 위한 미들웨어 구조 등 위치추적, 보안, 응급상황에 대처와 관련된 연구가 진행 중 이다^[4] 그러나, 가정이나 요양원, 헬스 센터 등에서 적용 가능한 미들웨어 연구는 미비하다^[5]. 따라서 본 논문에서는 사용자자동인식, 실시간 상태 모니터링에 적합한 센서중계노드를 설계하고 구현하였다.

2. SRN 기능

최근에는 센서네트워크에 대한 높은 신뢰도를 바탕으로 실시간으로 교통정보를 수집하여 차량사고를 감지하고 차량사고의 예측까지도 수행할 수 있는 모델도 새로이 제시되고 있다. USN 응용이 요구하는 QoS 조건을 만족시키는 것을 최우선 과제로 하여 만들어진 MiLAN 미들웨어, USN 응용의 질의를 센싱 데이터에 대한 특정 이벤트 조건으로 간주하여 처리하는 것을 주목적으로 하는 DSWare 미들웨어, USN 응용 서비스 모델의 상황 변화에 따라서 센서 노드의 기능을 무선 네트워크를 통하여 동적으로 갱신시킬 수 있는 것을 목적으로 하는 Impala 미들웨어, 센서 네트워크에 존재하는 센싱 데이터들을 분산 데이터로 간주하여 응용의 질의를 분산 데이터베이스에서의 질의 수행 과정으로 처리하는 것을 목적으로 하는 TinyDB, Cougar 미들웨어 등이 개발되어 왔다^[6].



(그림 1) SRN 기본 구조도

SRN은 생체정보 측정값들을 수집하는 센서노드들을 관리하고 어플리케이션과의 통신을 담당한다.



(그림 2) SRN 기능도

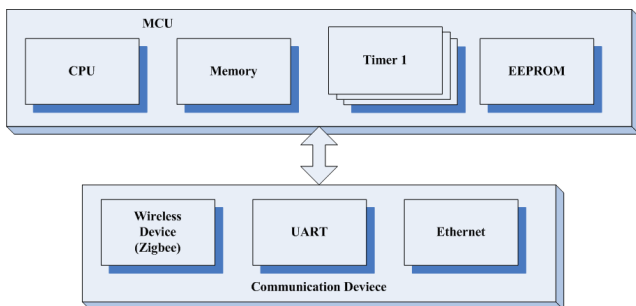
SRN은 서비스관리기능, 질의처리기능, 상황인식기능, 센싱데이터 관리기능, 외부 DB연계기능, 제어처리 기능, 베이스 스테이션 관리 기능으로 구성된다.

서비스 관리 기능은 생체정보측정 장치들과 연계되는 응용 서비스들에 대한 세션 정보를 관리하고 QoS 요구사항을 분석한다. 질의 처리 기능은 일시성/연속성/이벤트 형태의 질의들을 Ready, Running, Suspended Queue 를 이용하여 저장 및 관리를 수행한다. 상황인식 기능은 실시간 수집된 센싱 정보와 기존 데이터를 이용하여 새로운 건강 상황 정보를 생성한다. 제어 처리 기능은 응용 서비스의 질의에 대한 제어 명령을 수행하고, 센서 네트워크

및 센서 노드의 상태에 대하여 동적/정적 메타 정보를 관리한다. 센싱 데이터 관리 컴포넌트는 센서 네트워크로부터 실시간으로 제공되는 센싱 정보에 대한 캐싱 및 로깅 정보를 관리하고 응용서비스에 제공한다. 외부 DB 연계 기능은 상황인식 처리를 위하여 응용서비스에서 제공하는 외부 DB 를 연계하기 위한 역할을 수행한다. 베이스 스테이션 관리 기능은 생체정보측정기와 연계된 센서들이 동시에 연결 가능하도록 각 센서 네트워크의 게이트웨이 기능을 수행한다.

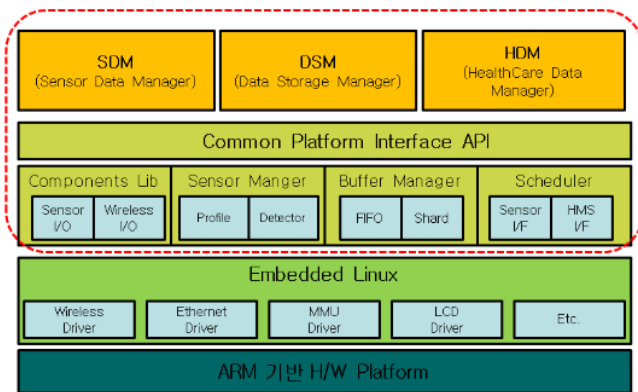
3. SRN 설계

3.1 SRN API 설계



(그림 3) SRN 시스템 구조

SRN은 임베디드 리눅스 환경에서 개발하여 센서노드로부터 획득한 생체측정정보를 받거나, SRN이 센서노드에 데이터 전송을 보낸다.



(그림 4) SRN API 설계

SRN API는 그림 4와 같이 SDM(Sensor Data Manager) API, DSM(Data Storage Manager), HDM(Healthcare Data Manager) API, Components Library, Sensor Manager, Buffer Manager, Scheduler로 구성된다.

SRN API는 통신 영역에 접속 대기 중이거나 접속된 SN를 자동 탐지하고, 센서 프로파일을 매핑 하도록 하였다. 센서노드에 관한 정보를 확보하고 Scheduler를 통하여 센서노드로 부터 실시간이나, 한 시간 단위, 하루 단위 등

센서 정보 빈도수에 맞추어 스케줄링 하였으며, 데이터관리를 위해 포터블 센서의 정보를 캐쉬하고, 응용 서비스로부터 분석한 결과를 포터블 센서에 전달하며, 전달시 센서가 접속되어 있지 않을 경우 정보를 저장하고 캐쉬된 정보가 너무 오래된 정보이면 타이머를 활용하여 삭제하도록 하였다

시스템 규모에 따른 처리를 분리하여 수행 가능한 환경을 제공 SRN은 여러 개의 Sensor Node와 Zigbee 통신하고, 동시에 Ethernet을 통하여 서비스를 제공한다.

3.2 통신프로토콜 설계

센서노드는 생체정보 측정기 종류인식, 생체정보취득, 취득 데이터 저장, 데이터 전송, 위험요소에 대한 경고 메시지 표시, 휴대장치 ID 및 사용자 ID 관리 기능 등을 전송할 수 있고 이러한 정보를 수집 관리하여 Application에 제공한다. 또한 Sensor Node에게 알람정보를 제공한다.

모 들		기 능	비 교
통신 모듈	Connection 관리	socket 연결 관리	tcp
		zigbee 연결 관리	Serial 통신
	어플리케이션과 통신	자체 프로토콜 송수신, CRC검증	
		센서노드 통신	자체 프로토콜 송수신, CRC검증
오류 관리	error처리	3회 이후 버림	
데이터	프로토콜 분석	자체 프로토콜 분석	정보분석
	프로토콜 생성	송신 메시지 생성	
분류	통신모듈 연계	통신모듈과 상호 데이터 교환	
	통신 event관리	송수신 데이터 처리를 위한 event발생	
	오류 관리	프로토콜 오류 처리	재전송 요구

- Application service와의 Message

구분	User_ID	SRN_ID	Time	SRN IP	SRN Port	Data Cnt	DATA		
							Attr	D_type	Value
길이	string (8byte)	string (12byte)	srtng (14byte)	string (15byte)	integer (2byte)	integer (2byte)	char (1byte)	char (1byte)	2byte~

User_ID : 사용자 식별코드(8Byte)-> 사전 등록된 고객의 식별 Code 로서 HMS 에서 생성하여 부여됨.
 향후 확장성을 고려하여 8자리로 정의하며, 4자리 입력 시 앞의 4자리는 "0"로 채워 8자리로 만듦

SRN_ID : SRN 식별코드(12Byte) ->제품번호

Time : 측정시간(14Byte-String)

SRN IP : SRN IP

SRN Port : Listen Port Number

Data Cnt : DATA 의 개수

DATA : Data는 (attr, d_len,d_type,Value)의 쌍으로 구성되며, "Data Cnt"의 개수 만큼 반복한다.

Attr's : Data 속성 Table 참조 (1Byte)

d_len : Value의 길이 (1Byte)

D_type : Value의 Data 속성-> i = integer , s = string

Value : 실 Data 값

- Sensor Node 와의 Message

구분	항목	크기
송신	Sensor Node ID	1Byte
	User ID	8byte
	측정 Sensor 데이터	2Byte
	측정 Sensor Kind	1Byte
수신	Sensor Node ID	1Byte
	User ID	8Byte
	Alarm Code	1Byte

4. SRN API 구현

```

■ SensorIO API
int initSerial(const char *portName)
- 해당 포트를 초기화한 후 연결하고 serialID를 리턴한다.
char *getDataSerial(int serialID)
- 연결된 포트로부터 데이터를 가져온다.
- return : -1 / 데이터 수신 실패
int sendDataSerial(char *str, int serialID)
- 연결된 포트에 데이터를 전송한다.
- return : 1 - 데이터 전송 성공, 0 - 데이터 전송 실패

int cleanupSerial(int serialID)
- 해당 포트를 연결을 해제한다.
- return : 1 - 연결해제 성공, 0 - 연결해제 실패

int isConnectedSocket(int serialID)
- 연결 serialID를 파라미터로 받아 연결 상태가 Connect 상태이면 1, Disconnect 상태이면 0을 리턴한다.

■ Wireless_Module API
int initServerSocket(int port)
- TCP 서버포트를 열고 Client 접속상태를 초기화하고 대기한다.
- return : 0 - 대기상태, 1 - 접속상태, -1 - 에러상태

int initClientSocket(int port)
- TCP 포트를 열고 Client 접속상태를 초기화한다.
- return : 1 - 접속성공, -1 - 에러상태

int ConnectionSocket(int port)
- TCP 포트를 통해 서버에 접속하고 접속상태와 ConnectionID를 리턴한다.
- return : ConnectionID - 접속성공, -1 - 에러상태

int isConnectedSocket(int ConnectID)
- 연결 ConnectID를 파라미터로 받아 연결 상태가 Connect 상태이면 0, Disconnect 상태이면 1을 리턴한다.

■ DAS_Management API
void RebootSystem()
- 모든 프로그램을 종료하고 시스템을 리부팅 한다.

void initSystem()
- DAS 시스템을 시작하고 초기화 한다.

■ Data_St API
int saveData(sensingData getdata)
- 센싱된 정보를 userid 별로 분류하여 저장한다.
- return : 1 - 성공, 0 실패

sensingData* loadData(char *userid)
- userid 별로 데이터를 로드한다.

■ Data_TxRx_Scheduler API
int sendMsgToHMS(char *message)
- message를 HMS로 보낸다.
- return : 1 - 성공, 0 - 실패

char *getMsgFromHMS()
- HMS로부터 메시지를 요청한다.

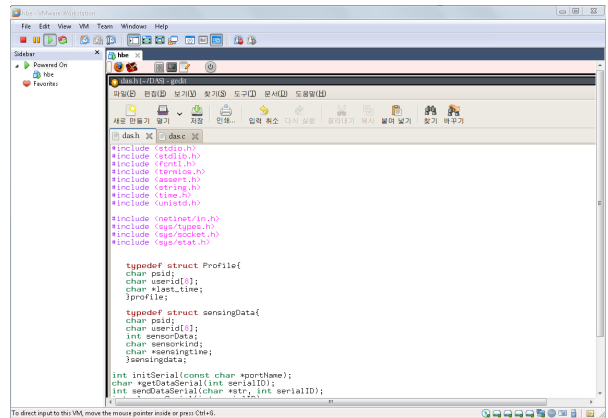
■ FIFO API
int is_empty(QueueType *q)
- 메시지 큐의 공백상태 검출한다.
- return : -1 - 공백상태

int is_full(QueueType *q)
- 메시지 큐의 포화상태 검출한다.

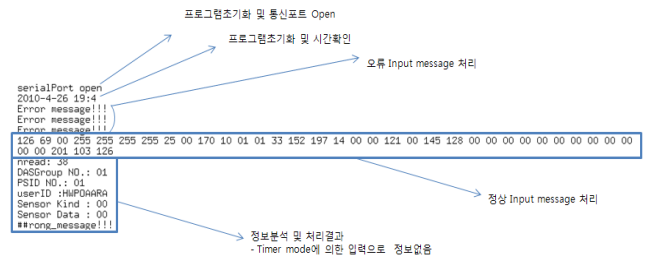
void enqueue(QueueType *q, element item)
- 메시지 큐에 데이터를 삽입한다.

element queue(QueueType *q)
- 메시지 큐의 내용을 전송한다.
typedef struct char *elemnet;
    
```

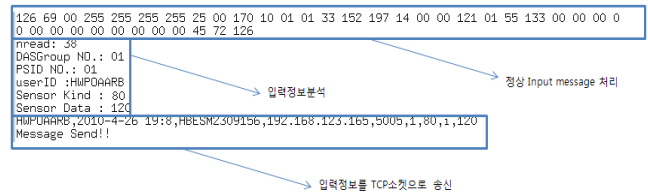
SRN API는 VMware 환경에서 C언어로 구현하여 gcc 컴파일러를 이용해 컴파일 한 후 SRN장비에서 실험하였다.



(그림 5) VMware 환경에서의 API 구현



(그림 6) SRN API를 활용한 결과



(그림 7) SRN API를 활용한 결과

그림 6과 7은 Sensor Node로부터 수집된 데이터를 어플리케이션에게 제공하는 모니터링 화면이다.

5. 결론

U-health 서비스를 제공하기 위해 휴대성과 이식성을 갖춘 생체정보 수집을 위한 센서노드의 정보를 수집하고 관리하는 SRN(Sensor Relay Node)를 설계 구현하였다. SRN은 서비스관리기능, 질의처리기능, 상황인식기능, 센싱데이터 관리기능, 외부 DB연계기능, 제어처리 기능, 베이스 스테이션 관리 기능으로 구성되며 된다.

센서노드로부터 생체정보 측정기 종류인식, 생체정보취득, 취득 데이터 저장, 데이터 전송, 위험요소에 대한 경고 메시지 표시, 휴대장치 ID 및 사용자 ID 관리 기능 등을 수집하고 이러한 정보를 수집 관리하여 Application에 제

공한다. 또한 Sensor Node를 통해 알람기능을 제공할 수 있다.

SRN의 구현을 통해 센서노드를 통한 생체정보 수집 및 처리에 대한 방안을 제안하였다. 향후 다양한 서비스에 제공할 수 있도록 Application의 개발과 생체정보를 전송하는 센서노드 정보 전송방식의 표준화를 통해 좀 더 다양한 정보를 광범위하게 수집하고 활용할 수 있도록 하는 연구가 이루어져야 할 것으로 보여진다.

참고문헌

- [1] 최보경, 윤희용 “상황인식 기반의U-Silvercare 서비스”, 정보과학회논문지, 제 36권 제 3호, 2009.3, pp200-207.
- [2] Weiwei Yuan, Donghai Guan, Sungyoung Lee,heeijo Lee, Using Reputation System in Ubiquitous Healthcare, e-Health Networking,Application and Services, 182-186, 2007
- [3] Munoz, M. A., Rodriguez, M., Favela, J.,Marinez-Garcia A. I., Gonzalez, V.M,Context-aware mobile communication inhospitals. Computer IEEE, 36, 9, 38-46, 2003
- [4] Jakob E. Bardram, Henrik Baerbak Christensen,Middleware for Pervasive Healthcare, A WhitePaper
- [5] 김재열, 김용환, 안광선, “유헬스케어 시스템을 위한 적응형 미들웨어” 한국정보과학회 학술발표논문집 Vol.34, No.2(B). 2007
- [6] 김찬호, “유비쿼터스 서비스와 아키텍처에 관한 고찰 : 심장병 환자를 위한 건강관리 시스템을 중심으로” 2008. 한국과학기술원, 석사학위논문