

## ATmega128 기반 단문 메시지 전송 프로토콜 IMCP 구현★

김 점 구\*

### 요 약

사회관계망서비스(SNS)는 무료이지만 통신사에 지불하는 데이터 이용요금과 회원의 정보는 직·간접으로 제공이 되어야 한다. 그리고 SNS는 스마트 폰과 PC 등 송·수신 기기의 사양이 날로 높아가는 반면에 주로 단문 메시지를 이용하여 작업지시 등을 하는 오염지역이나 반도체 제조공장 등 특수 환경에서 범용 전송 프로토콜을 이용하는 것은 기기조작이 자유롭지 못하고, 보안에 취약한 문제점을 가지고 있다. 본 논문은 위험과 오염 그리고 크린 존(clean zone)등 특수 환경에서 웨어러블을 지향하는 저전력 저비용의 메시지 전송 프로토콜 IMCP를 ATmega128 기반으로 구현하여 동작 시험 통하여 실용성 검증하였다.

## Implementation of ATmega128 based Short Message Transmission Protocol IMCP

Kim Jeom Goo\*

### ABSTRACT

The social networking service (SNS) is free, but the data usage fee paid to the telecommunications company and the member's information must be provided directly or indirectly. In addition, while SNS' specifications for transmitting and receiving devices such as smart-phones and PCs are increasing day by day, using universal transmission protocols in special environments such as contaminated areas or semiconductor manufacturing plants where work instructions are mainly made using short messages is not easy. It is not free and has a problem of weak security. This paper verified the practicality through the operation test by implementing IMCP, a low-power, low-cost message transmission protocol that aims to be wearable in special environments such as risk, pollution, and clean zone based on ATmega128.

**Key-words:** SNS, IMCP, protocol, special environments, short messages

접수일(2020년 7월 13일), 게재확정일(2020년 9월 30일)

\* 남서울대학교 컴퓨터소프트웨어학과 교수

★ 본 논문은 2019학년도 남서울대학교 교내연구비 지원에 의해서 연구되었음.

## 1. 서 론

정보통신기술의 발전은 다양한 형태의 사회관계망 서비스(SNS : Social Network Service)가 만들어지는 기술적 기반이 되었다. 즉, 문자, 사진, 동영상 등 다양한 종류의 데이터를 서로 주고받으면서 사회적인 중요한 소통과 공동체 활동의 매체 역할을 하고 있다[1].

이러한 SNS는 사회 전반적인 분야에서 중요한 소통의 기능을 하고 있고, 심지어 사이버 여론을 형성하는 역할도 함으로써 신문과 방송에 이은 매우 중요한 공공 매체로 자리 매김을 했다고 여겨진다. 그리고 더 나아가 새로운 형태의 소통 수단과 흥미를 가미한 콘텐츠를 개발하여 그 기능과 역할을 확대해 가고 있다. 반면에 이러한 SNS의 기능을 실행하는 기기인 스마트폰이나 PC 등은 점차 높은 사양이 요구되고, 이를 이용하는 네트워크 사용료와 데이터 이용요금 역시 사용자에게는 적지 않은 부담이 되고 있다.

그럼에도 이러한 스마트폰과 PC기반 SNS는 산업현장의 위험지역이나 전염병 등 오염지역, 그리고 반도체 제조공장 등과 같이 보호장구와 방진복장을 갖추어야 하는 특수 환경에서 직접 메시지 입력에 어려움이 있는 경우 SNS를 이용하여 의사 전달을 하는 것은 매우 제한적일 수 있다. 그리고 이러한 특수 환경의 지협적인 공간에서 소통을 위해 정보통신망사업자(ISP)의 자원을 이용하고 이용 요금 등을 지불하는 것은 비효율적이다.

따라서 본 논문은 특수 환경에서 메시지를 이용한 개인과 개인 그리고 개인과 그룹간 작업 지시 등에 저전력과 저비용으로 효율적인 메시지를 전송할 수 있고, 웨어러블을 지향하는 ATmega128 기반 메시지 전송 프로토콜 IMCP(Intelligent Memo Control Protocol)를 구현하였으며 검증을 통하여 실용성을 입증하였다.

## 2. 관련연구

### 2.1 AVR ATmega128

AVR(Alf Vergard Risc)은 미국 아트멜(Atmel)사에서 만든 범용 RISC(Reduced Instruction Set

Computer) 마이크로 컨트롤러이다. 내부에 CPU, ROM, RAM, 플래시 메모리, ADC(Analog Digital Convertor), DAC(Digital Analog Convertor), GPIO(Gernel Purpose I/O)가 모두 내장되어 있다. PIC와 함께 독립된 상표 형식으로는 8비트 마이크로 컨트롤러들 중에서 가장 많이 알려져 있다. 특히 ISP(In circuit Serial Programming) 인터페이스를 갖추고 있어서 별도의 롬 라이터 없이 PC에서 프린터 포트나 시리얼 포트, USB 인터페이스에 연결하는 ISP 장비를 통해 손쉽게 프로그램을 작성하여 탑재할 수 있어서 편리하다[2][9].

저렴한 가격에 ROM, RAM 걱정 없이 프로그램을 탑재할 수 있고, 8비트급 MCU(Micro Controller Unit)임에도 RISC를 표방하고 있어 명령어 길이가 일정하고 범용 레지스터 숫자가 많으며, 하버드 아키텍처가 도입되어 메모리 공간이 비교적 일정하게 구성되어 있다는 RISC적인 특징을 가지고 있다[2].

AVR은 tiny, mega, xmega 등 여러 종류가 있으며, 본 논문에서는 8 - 256Kb 프로그램 메모리 28 - 100핀 패키지 확장된 명령어 집합(곱셈 명령어, 큰 프로그램 메모리를 처리를 위한 명령어), 확장된 주변 장치 세트 등의 특징을 지닌 ATmega128을 이용한다. 주변장치로는 SPI, I2C, PWM, ADC 등이 있으며, Timer/Counter, 전원 관리 등의 기능이 있어 기능 추가도 가능하다[10][11][12].

### 2.2 IEEE 802.11b 무선통신 프로토콜

IEEE 802.11은 현재 주로 쓰이는 유선 LAN 형태인 이더넷의 단점을 보완하기 위해 고안된 기술로, 이더넷 네트워크의 말단에 위치해 필요 없는 배선 작업과 유지관리 비용을 최소화하기 위해 널리 쓰이고 있다. 보통 폐쇄되지 않은 넓은 공간에 하나의 핫스팟을 설치하며, 외부 WAN과 백본 스위치 각 사무실 핫스팟 사이를 이더넷 네트워크로 연결하고, 핫스팟부터 각 사무실의 컴퓨터는 무선으로 연결함으로써 사무실 내에 번거로이 케이블을 설치하고 유지보수를 하지 않아도 된다[6].

802.11은 최고속도가 2Mbps인 무선 네트워크 기술로, 적외선 신호나 ISM(Industry Science Medical) 대역인 2.4GHz 대역 전파를 사용해 데이터

를 주고받으며, 여러 기기가 함께 네트워크에 참여할 수 있도록 CSMA/CA 기술을 사용한다. 그러나 규격이 엄격하게 정해지지 않아서 서로 다른 회사에서 만들어진 802.11 제품 사이에 호환성이 부족했고 속도가 느려서 널리 사용되지 않았다.

802.11b는 802.11 규격을 기반으로 더욱 발전시킨 기술로, 최고 전송속도는 11Mbps이나 실제로는 CSMA/CA 기술의 구현 과정에서 6~7Mbps 정도의 효율을 나타내는 것으로 알려져 있다[28].

### 2.3 UART(범용 비동기화 송수신기)

UART(Universal Asynchronous Receiver Transmitter)는 병렬 데이터의 형태를 직렬 방식으로 전환하여 데이터를 전송하는 컴퓨터 하드웨어의 일종이다. UART는 일반적으로 EIA RS-232, RS-422, RS-485와 같은 통신 표준과 함께 사용한다. UART의 U는 범용을 가리키는데 이는 자료 형태나 전송 속도를 직접 구성할 수 있고, 실제 전기 신호 수준과 방식이 일반적으로 UART 바깥의 특정 한 드라이버 회로를 통해 관리를 받는다. 통신 데이터는 메모리 또는 레지스터에 들어 있어 이것을 차례대로 읽어 직렬화 하여 통신한다. 최대 8비트가 기본 단위이다[5].

UART는 비동기 통신이므로 동기 신호가 전달되지 않는다. 따라서 수신 쪽에서 동기신호를 찾아내어 데이터의 시작과 끝을 시간적으로 알아 처리할 수 있도록 약속되어 있다. 디지털 회로는 자체의 클럭 신호를 추가로 사용하여 정해진 속도로 수신 데이터로 부터 비트 구간을 구분하고 그 비트의 논리 상태를 결정하여 데이터 통신을 하는 USRT(Universal Synchronous Receiver Transmitter)도 사용한다.

UART는 보통 마이크로컨트롤러에도 포함되어 있다. 듀얼 UART, 곧 DUART는 두 개의 UART를 하나의 칩에 합친 것이다. 수많은 현대의 집적 회로(IC)는 동기화 통신인 USRT도 함께 지원한다.

## 3. ATmega128 기반 단문 메시지 전송 프로토콜 IMCP 구현

### 3.1 IMCP 전송 프로토콜

메시지 단말기와 메시지 서버간의 데이터 통신을 위하여 정확하게 메시지 제어 및 전달을 할 수 있는 프로토콜이 필요하다.

본 논문에서 요구되는 프로토콜은 신뢰성 있는 메시지 전송으로 정확히 메시지를 처리할 수 있어야 하며, 제한된 시스템 자원에 따라서 적은 자원을 필요로 하는 프로토콜이 필요하다. 또한, 흐름 제어 및 오류 감지 기능을 수행하고 재전송할 수 있어야 하며, 각 메시지 단말기의 상태를 확인하고 관리할 수 있어야 한다. 정확한 데이터 통신을 위하여 본 논문에서는 메시지를 Message DESCRIPTION, Message HEADER, Message BODY로 나뉘서 단문 메시지 전송 프로토콜 IMCP를 정의하고 구현하였다.

#### 3.1.1 Message DESCRIPTION

Message DESCRIPTION은 전송할 전체 메시지의 간단한 설명 구문이다. 전송 메시지가 어떠한 역할을 수행하는지를 정의한다. 메시지의 종류에는 보고, 완료, 재전송 등 전체 메시지의 목적에 대해서 정의하고 메시지의 목적을 기술한다.

Message DESCRIPTION에 따라서 Message HEADER의 내용이 변경되고, Message DESCRIPTION에는 프로토콜의 VERSION이 정의되어 있어 향후 다른 VERSION의 프로토콜을 해석할 때는 반드시 해당 VERSION의 정의대로 해석하여야 한다. Message DESCRIPTION은 다음과 같은 구문(syntax)을 갖는다.

```
METHOD NAME VERSION\r\n
```

METHOD는 전체 메시지에 대한 간단한 기술 구문이며, 본 논문의 프로토콜 명칭인 IMCP로 명명하였다. VERSION은 현재 개발 버전으로 하고, METHOD는 <표 1>과 같이 정의한다.

<표 1> Message DESCRIPTION METHOD

구 문	의 미	메시지 방향
REPORT	<ul style="list-style-type: none"> <li>- 메시지 단말기에서 구성한 화면을 전체적으로 종합하여 메시지 서버에게 전달.</li> <li>- 메시지 길이가 반드시 정의되어야 함.</li> </ul>	C → S
RE-REPORT	<ul style="list-style-type: none"> <li>- 메시지 서버가 전송받은 메시지에 오류가 있어 재전송을 요청.</li> <li>- 메시지 단말기는 즉시 재전송하여야 함.</li> </ul>	S → C
ACK	<ul style="list-style-type: none"> <li>- 정상 수신 알림.</li> </ul>	S → C C → S
DISREU-TION	<ul style="list-style-type: none"> <li>- 메시지 서버에서 수신한 메시지를 메시지 단말기에 배포.</li> <li>- 메시지 단말기는 수신 후 즉시 ACK을 전송하여야 함.</li> </ul>	S → C
REQUEST	<ul style="list-style-type: none"> <li>- 메시지 단말기가 메시지 서버에게 최근 메시지 요청.</li> <li>- 메시지 단말기는 전원 인가 후 바로 요청을 하여야 함.</li> </ul>	C → S

※ C-메시지 단말기, S-메시지 서버

### 3.1.2 Message HEADER

Message HEADER에는 데이터 전송 및 상태 제어를 위한 제어 코드 작성과 실제 전송되는 데이터 Message BODY에 기술한다.

Message HEADER와 Message BODY는 연속된 "\r\n" 문자로써 구분되며, Message BODY의 총 길이는 Message HEADER에 정의되어 있어 전송 오류 탐지 및 데이터의 종단을 알 수 있도록 한다.

Message HEADER에는 데이터 전송 흐름 제어, 오류탐지, 클라이언트 상태 제어 등을 정의하며 <표 2>와 같은 구문과 의미(semantics)를 갖는다.

<표 2> Message HEADER 정의

구 문	의 미	메시지 방향
FROM	<ul style="list-style-type: none"> <li>- 메시지 발신 장치의 이름을 나타냄.</li> </ul>	S → C C → S
TO	<ul style="list-style-type: none"> <li>- 메시지 수신 대상을 나타냄.</li> <li>- ALL은 전체 기기를 의미함.</li> </ul>	S → C C → S
LENGTH	<ul style="list-style-type: none"> <li>- 메시지 길이.</li> <li>- Message BODY만의 길이를 의미함.</li> </ul>	S → C C → S
DATE/TI-ME	<ul style="list-style-type: none"> <li>- 메시지의 생성 날짜/시간을 의미함.</li> <li>- YYYY/MM/DD HH/MM/SS로 표시하며 시간은 24시간으로 표시함.</li> </ul>	S → C C → S

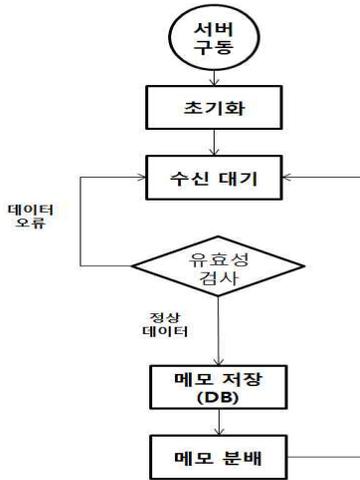
### 3.1.3 Message BODY

Message BODY는 메시지 단말기의 화면을 구성할 수 있는 데이터로 구성된 2진 바이너리 형태이며, 사용자가 메시지 단말기에 입력한 화면의 메시지를 의미한다. 이 메시지는 메시지 단말기에서 구성하며 메시지 서버로 전달한다. 메시지의 끝은 특별히 명시하지 않고 Message HEADER의 LENGTH를 이용하여서 계산한다. 메시지는 구분되지 않은 단일 메시지로 전송한다. 전송되는 메시지 영역의 전체 크기는 348 x 158 픽셀이며, 8개의 컬러 메시지로 구성된다.

## 3.2 메시지 서버

### 3.2.1 메시지 서버의 제어

메시지 서버는 메시지 단말기들이 전송한 메시지를 저장하고 관리하며, 메시지를 배포하는 역할을 담당한다. 메시지 단말기들로부터 전송받은 메시지는 유효성 검사 후 즉시 데이터베이스로 저장되어야 하며, 메시지 단말기들에게 전송한다. (그림 1)은 메시지 서버의 동작을 나타내며, 메시지 서버의 구현 환경은 <표 3>과 같다.



(그림 1) 메시지 서버의 동작 흐름

<표 3> 메시지 서버의 구현 환경

항 목	특 징
CPU	Intel i5
RAM	2G
HDD	32G
OS	Linux (Ubuntu 3.5.0-23-generic)
Compiler	gcc version 4.6.3
Database	MySQL version 5.5.29

메시지 저장을 위한 데이터베이스는 MySQL을 사용하며, <표 4>와 같은 데이터 스키마를 사용한다.

<표 4> 메시지 서버 메시지 저장 스키마

Field	Type	Null	Key	Default	Extra
no	int(5)	no	PRI	Null	auto_increment
client_name	var char (32)	no		Null	
datetime	datetime	no		Null	
memo	mediumblob	no		Null	

저장되는 데이터의 고유성을 유지하기 위하여 n

o를 기본 키로 설정하고 auto\_increment 속성을 추가한다. 메시지 생성 단말기는 client\_name에 저장되며, datetime에는 메시지 저장 일자와 시간이 저장되며, 메시지는 memo 필드에 2진 데이터인 mediumblob 형식으로 저장된다. 메시지 서버의 주요 제어 알고리즘은 (그림 2)와 같다.

```

while(1)
{
    client = wait_client();
    parse_message(client);

    if( validate_test(client) == VALIDATE_ERROR)
    {
        continue;
    }

    store_memo(client);

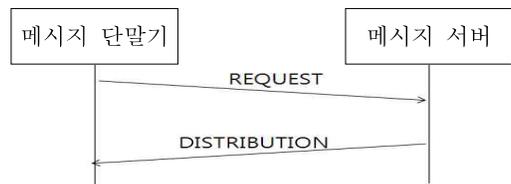
    distribute_memo();
}
    
```

(그림 2) 메시지 서버의 주요 제어 알고리즘

### 3.2.2 메시지 서버의 기능

#### 가. 메시지 획득과 분배

(그림 3)과 같이 메시지 단말기가 메시지 서버에게 최신의 메시지를 요구하고 메시지 서버는 메시지 단말기에 메시지를 분배한다.



(그림 3) 메시지 획득과 분배

```

REQUEST IMCP 0.9\r\n
FROM: ITD_A\r\n
\r\n
    
```

메시지 단말기는 메시지 서버에 저장된 메시지를 획득하기 위하여 REQUEST 구문을 사용하여

FROM 헤더에 값을 넣은 다음 메시지 서버로 전달한다. FROM 헤더의 값은 메시지 단말기에 설정된 이름이다. 또한, 메시지 단말기는 수신 버퍼를 준비하고, 메시지 수신을 준비하는 단계로 전환된다.

```
DISTRIBUTION IMCP 0.9\r\n
FROM: ITD_C\r\n
TO: all\r\n
LENGTH: 11104\r\n
DATA/TIME: 2013-04-28 02:29:19\r\n
\r\n
DATA
```

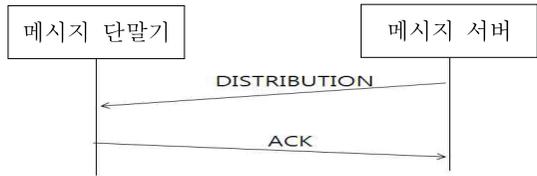
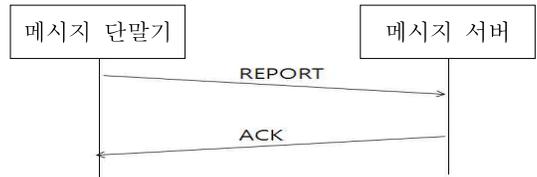
메시지 서버는 데이터베이스를 검색하여 가장 최근에 입력한 메시지를 선택하여 응답한다. FROM 헤더는 데이터베이스에 저장된 메시지 생성자이며, LENGTH는 DATA의 길이를 의미한다. TO 헤더는 메시지를 수신할 대상 메시지 단말기이며, 전체로 메시지를 배포할 때는 옵션 값을 ALL로 정의한다. DATA/TIME은 데이터베이스에 저장된 메시지 생성일이며, 저장된 메시지 데이터가 없을 경우는 다음과 같은 메시지가 전달된다.

```
DISTRIBUTION IMCP 0.9\r\n
FROM: 0\r\n
TO: 0\r\n
LENGTH: 0\r\n
DATA/TIME: 0\r\n
\r\n
```

모든 헤더 값은 0이며, 특히 BODY 부분에 데이터가 없다. 따라서 메시지 단말기는 수신된 HEAD의 LENGTH 값이 0이면 저장된 메시지가 없다는 것으로 해석하여야 한다.

**나. 메시지 저장과 분배**

(그림 4)와 같이 메시지 단말기가 작성한 메시지를 메시지 서버에게 전달하여 데이터베이스에 저장하고 분배한다.



(그림 4) 메시지 저장과 분배

```
REPORT IMCP 0.9\r\n
FROM: IUA_A\r\n
TO: ALL\r\n
LENGTH: 11104\r\n
\r\n
DATA
```

메시지 단말기는 REPORT 메소드를 사용하여 사용자가 작성한 메시지를 메시지 서버로 전달한다. 메시지 서버는 메시지를 DB에 저장 후 ACK을 전송한다.

```
ACK IMCP 0.9
FROM: IMCP SERVER
TO: IUA_A
```

ACK 전송 이후에 메시지 단말기들에게 메시지를 DISTRIBUTION 구문을 이용하여서 배포한다.

```
DISTRIBUTION IMCP 0.9\r\n
FROM: ITD_A\r\n
TO: all\r\n
LENGTH: 11104\r\n
DATA/TIME: 2013-04-28 02:35:4\r\n
\r\n
DATA
```

FROM 헤더 값은 메시지를 작성한 메시지 단말기의 이름이며 LENGTH는 DATA의 길이를 나타낸다. DATA/TIME 헤더 값은 메시지를 수신한 날짜/시간 값으로 메시지 서버가 메시지 데이터를 수신할 때 시간을 계산하여 데이터베이스에 저장한다.

### 3.3 메시지 단말기

#### 3.3.1 메시지 단말기의 제어

메시지 단말기의 모든 제어는 ATmega128이 담당하며, 각종 데이터를 처리하고 사용자 인터페이스를 제공한다. 또한, TFT-LCD의 데이터를 제어하며, 사용자가 입력한 메시지를 서버로 전송하는 제어를 담당한다. 메시지 단말기의 기능은 <표 5>와 같다.

<표 5> 메시지 단말기 기능

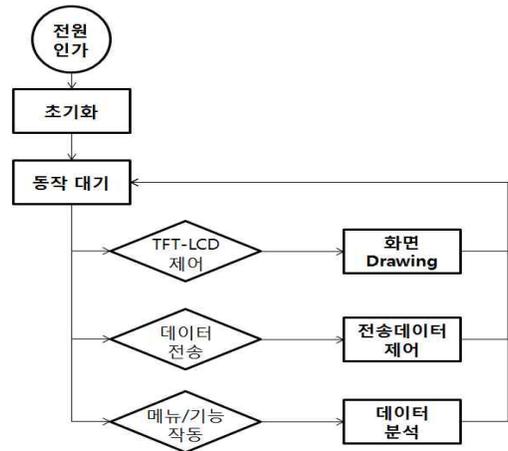
제어 대상		기능
H/W	TFT-LCD	2.6인치 TFT-LCD 화면 출력
	Touch Position	I2C 제어 기반의 Touch 위치
	Memo Data	TFT-LCD에 Drawing 된 데이터
	System	시스템 클럭 및 초기화
	GPIO	TFT-LCD Data 제어
S/W	TFT-LCD Drawing	TFT-LCD의 Drawing 계산
	Touch Position	Touch 위치 계산
	Memo Data	TFT-LCD에 Drawing 된 데이터
	Data Transmission	데이터 송수신
	Timer	시스템 시간

ATmega128은 전원 인가 후 시스템 타이머, GPIO, 클럭, I2C 등을 초기화한다. 이 과정에서 각

종 주변장치가 활성화되며, 초기값을 전송한다. TFT-LCD는 Data-sheet에 따라서 각종 전압 및 화면 구동 모드 등을 정의한다.

그리고 GPIO의 동작 및 시스템 타이머 등을 설정하여 시스템 전체 동작을 제어할 수 있도록 한다. 동작 중에는 GPIO 기반의 TFT-LCD 제어(i80 방식)를 하며, TFT-LCD 제어를 위한 RST, RD, RW 등을 제어하여 데이터를 전송한다. 또한, TFT-LCD에 기록된 메시지를 데이터로 변환하여 다시 TFT-LCD에 기록하면서 메뉴 기능 또한 동작하도록 제어한다.

그리고 서버로 메시지 전송을 수행하여 사용자가 전송 명령을 수행한다. 메시지 단말기 제어 동작은 (그림 5)와 같다.



(그림 5) 메시지 단말기 제어 동작 흐름

전원 인가 후 메시지 단말기는 각종 주변장치를 초기화하고 사용자의 메시지가 입력되기를 기다린다. 입력된 구문에 따라서 메뉴/기능 작동 분석에서 종류와 동작을 분석하고 정의하며, TFT-LCD Drawing이면 I2C 데이터 분석을 수행하여서 TFT-LCD의 화면을 제어하고, 데이터 전송이면 UART를 제어하여서 서버와의 데이터 송수신을 제어한다.

## 4. IMCP 검증

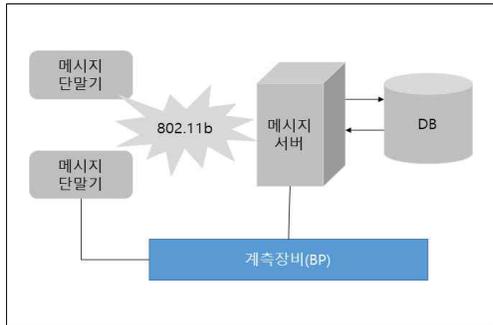
### 4.1 검증 환경

<표 6> 메시지 전송 시스템 성능 시험 장비

구 분	장 비	수 량	비 고
ICMP	메시지 서버	1 EA	대상 장비
	메시지 단말기	3 EA	
계측기	Breaking Point	1 EA	네트워크 장비 임대
통신 프로토콜	802.11b	1 EA	Fiber 2 ports/UT P 2 ports이상

시스템의 응답속도, 처리속도는 <표 6>과 같은 시험 장비를 연결하여 (그림 6)과 같이 시험 환경을 구축하고 네트워크 트래픽 계측 장비인 블랙킹 포인트(BP : Breaking Point)를 이용하여 성능 시험을 하였다.

(그림 6)은 계측장비인 BP와 메시지 서버 그리고 메시지 단말기를 구성하여 메시지 처리량 및 대기시간을 측정 할 수 있는 환경을 구축하였다.



(그림 6) IMCP 검증 환경

### 4.2 성능 평가

본 논문은 IMCP 서버의 응답속도, 메시지 처리 속도 등의 성능 평가 지표는 ISO15408 기준을 적용하여 평가를 하였다.

<표 7> 지연시간 시험 데이터 및 측정결과

Frame Size	Tx Count (frames)	Tx Frame Rate(fps)	Min Latency (μs)	Avg Latency (μs)	Max Latency (μs)
64	892857000	1488096	3340	3360	3380
64	892857000	1488096	3200	3200	3220
128	506757000	844595	5280	5200	5300
128	506757000	844595	4980	5200	5300
256	271739400	452899	8580	9140	9160
256	271739400	452899	9100	9140	9140
512	140977200	234952	16800	16800	16820
512	140977200	234952	15740	15740	15780
1024	71839200	119732	30100	32200	32200
1024	71839200	119732	32140	32180	32180

ICMP 단말기와 서버의 경우 패킷 처리 시 소요시간으로 대기편차를 인지 감각 상 늦다고 여겨지지 않을 정도인 2.5sec 이하를 목표로 하였으며, BP를 이용하여 10분간 테스트하여 응답속도를 측정하였다. <표 7>은 계측기와 시료를 각 패킷 길이(64, 128, 256, 1024 bit)로 나눠서 응답시간을 측정한 결과 평균 3,360~32,180μs까지 확인되었다.

이는 8bit의 프레임을 구성하고 있는 본 논문의 제안 시스템의 지연(대기)시간은 1.56sec로 시험 환경이 아닌 실 환경에서 노이즈 등을 고려해도 2.0sec 이내의 대기시간 요건을 갖는 데는 충분한 것으로 평가되었다.

## 5. 결론

산업현장의 위험한 환경과 전염병 검사 및 치료 시설 그리고 반도체 공장 등 특수 환경에서 메시지를 이용한 소통 수단으로 활용되는 전송기기는 기능의 다양성 보다는 방호복과 방진복을 착용한 상태에서도 의사 전달이 가능하도록 웨어러블 하거나 스티커처럼 탈부착이 용이하고 저전력 저비용의 효율성이 우선되어야 한다.

제안 IMCP 프로토콜은 ATmega128의 장점을 최대한 활용하였고 터치스크린을 이용한 필기중심의 메시지 구성과 터치방식의 명령어 수행으로 키 입력 방식을 지양함으로써 기기조작이 자유롭지

못한 환경에서도 쉽게 메시지를 전송할 수 있게 하였다.

IMCP 대한 검증은 기능 동작 후 데이터의 무결성과 처리속도, 그리고 최대 세션 연결과 유지를 실험결과 유사한 시험환경을 구축하여 시험하였다. 그리고 공인 계측기 BP를 이용하여 성능시험을 통해 목표치 보다 우수한 결과 값을 획득함으로써 입증하였다.

본 논문에서 제안하는 IMCP를 이용한 메시지 서버는 일반 PC 수준이면 가능하고, 메시지 단말기는 ATmega128를 활용함으로써 적은 비용으로 구축이 가능하고, 메시지 단말기의 전력 소모가 매우 낮아 효율적이고, SNS 환경에서는 개인이 여러대의 전송기기를 보유하기 어려웠던 것을 개인의 활동 범위 내에서 팔이나 손목 등에 착용할 수 있고, 종이 스티커처럼 메시지를 부착하듯 자유롭게 작업 환경을 이동하면서 메시지를 작성하고 전송할 수 있는 편리함을 갖게 될 것이다.

메시지 단말기의 개수는 제한이 없으며, 필요한 만큼 확장 가능하다. 향후 IMCP를 기반으로 한 서버와 단말기를 구현하여 실증 연구가 필요하다.

## 참고문헌

- [1] 신호영, “SNS 이용 동기에 따른 이용행위가 SNS 만족감에 미치는 영향 연구”, 연세대학교 정보통신대학원 석사논문, 2016. 02
- [2] 이중재, “AVR 마이크로프로세서와 그래픽 LCD를 이용한 회전도립진자 시스템 제작”, 선문대학교 대학원 석사논문, 2006. 02
- [3] 서영호, “칩 테스트를 위한 UART-to-APB 인터페이스 회로 설계”, 한국항행학회, 2019. 08
- [4] 윤국재, “IEEE 802.11b 무선랜에서 에너지 절약을 위한 스테이션의 전력 제어”, 호서대학교 대학원 박사논문, 2008. 02
- [5] A. Isizo. h, “Throughput Analysis of IEEE 802.11b Wireless Lan with One Access Point using Opnet Simulator”, IJASA, 2012. 03.
- [6] Behrouz Forouzan, “Data Communications and Networking”, McGraw-Hill, 2003. 08.

- [7] John Catsoulis, “Designing Embedded Hardware”, OReilly, 2002. 11.
- [8] JIM LEDIN, “Embedded Control Systems in C/C++”, CMP, 2004. 01.
- [9] T. Sridhar, “Designing Embedded Communications Software”, CMP, 2003. 07.
- [10] TAMMY NOERGAARD, “EMBEDDED SYSTEMS ARCHITECTURE”, NEWNES, 2005. 06.
- [11] Paul DuBois, “MySQL”, SAMS, 2008. 06.
- [12] Bradford, Ronald, “Effective MySQL”, McGraw-Hill, 2012. 02.
- [13] MSN Messenger, <http://windowlive.joinsmsn.com/im/wl/main.asp>
- [14] AIM, <http://www.aim.com/>
- [15] Atmel, <http://www.atmel.com/>
- [16] MySQL, <http://www.mysql.com/>

## [ 저자 소개 ]



김 점 구 (Jeom Goo Kim)  
 1990년 2월 광운대학교  
 전자계산학과 이학사  
 1997년 8월 광운대학교  
 전자계산학과 석사  
 2000년 8월 한남대학교  
 컴퓨터공학 박사  
 1999년 3월~ 현재  
 남서울대학교  
 컴퓨터소프트웨어학과  
 교수  
 email : jgoo@nsu.ac.kr