

# 무선 네트워크를 이용한 임베디드 다중 전광판 시스템의 구현

정도욱\*, 정재훈\*, 한형철\*, 차정운\*\*, 김영곤\*

\*한국산업기술대학교 컴퓨터공학과

\*\*㈜오리엔탈컨설팅

e-mail: moderato@kpu.ac.kr

## Implementation of the Embedded Multi-Billboard System using Wireless Network

Do-Wook Jeong\*, Jae-Hun Jung\*, Hyeong-Cheol Han\*,  
Jeong-Un Cha\*\*, Young-Gon Kim\*

\*Dept of Computer Engineering, Korea Polytechnic University

\*\*Orientalconsulting Co.,Ltd

### 요 약

본 논문에서는 WiFi 환경이 제공되는 요즘의 환경에 맞추어 전광판을 무선 네트워크망으로 관리하기 위한 시스템을 제안하려고 한다. ATmega2560코어와 WiFi 모듈 및 서버를 활용하여 제작한 본 시스템은 다양한 출력이 불가능한 기존의 전광판과 전광판에 필요한 임베디드 코어에 문자를 출력하는데 필요한 다양한 변환과정으로 생긴 부하를 해결하는데 초점을 맞추었다. 이 시스템은 크게 두 가지의 장점을 지닌다. 한 가지는 사용자에게 다중 전광판의 출력 내용을 직접 입력할 필요 없이 서버를 통하여 쉽게 관리할 수 있다는 점이다. 다른 하나는 전광판 관리자에게 출력할 폰트나 출력 양식을 서버로 집중 관리하게 함으로서 쉽게 변동할 수 있게 한다는 점이다.

### 1. 서론

무선 통신 시장의 발전으로 현재 WiFi가 사용되는 범위가 크게 확대되었다. 따라서 인터넷 프로토콜에 의한 네트워크에 접속이 용이하며 이는 공장과 회사 내 인트라넷으로 이의 사용되는 폭이 넓어지게 하였다.

WiFi는 RF(Radio Frequency) 방식에 비하여 송수신거리가 짧으나 데이터의 고속 송수신이 가능하며, 일반적인 지그비나 블루투스에 비하여 송수신거리가 긴 장점이 있어 특히 공간의 격리가 이루어지지 않은 공장형태의 기업에 적합한 환경을 제공한다[1].

현재 국내 시장에 제공되는 LED 전광판의 경우 다양한 LED의 개발과 발전에 힘입어 대량생산되고 있으며 해외에 수출할 정도로 그 경쟁력이 증대되고 있다.

그러나 대부분의 전광판은 유선을 활용하는 시스템이 대부분이며 무선통신을 활용한 전광판의 경우 리모컨과 RF 통신을 활용한 전광판 시스템으로 한정지어져 있다. 또한 기업 프로모션으로 활용되는 대형 전광판의 경우 이동통신사의 SMSC(Short Message Service Center)를 통하여 잠재고객이 보낸 SMS(Short Message Service)가 전광판으로의 출력을 가능하게 한 시스템으로 한정된다[2].

이는 공장이나 회사에서 서로간의 효율적인 미션 이해를 위한 전광판의 활용과 작은 사업체 내에서 광고와 게시판을 목적으로 하는 전광판, 그리고 개인적으로 PR의 목적으로 활용하는 전광판의 컨트롤에는 부적절하다.

따라서 WiFi를 통하여 네트워크를 활용한 전광판을 제안

한다. 이는 전광판의 사용자들이 기존의 전광판의 컨트롤이 그 근거리에서만 가능한 제약으로 인하여 겪는 문제를 해결 할 수 있다.

본 논문이 제안하는 방식은 Atmel사의 마이크로컨트롤 코어와 WiFi module을 활용하여 AP(Access Point)송수신 거리 하의 공간에서 IPv4(Internet Protocol version 4)를 통한 다중의 전광판을 제어 가능하도록 하였다. 임베디드 시스템으로 개발한 프로그램과 하드웨어는 그 하나로 다중의 LED 전광판을 제어 가능하도록 하였으며, 이는 대형의 전광판에도 5G WiFi의 발전으로 바로 응용가능하다.

또한 전광판에 출력되어야 할 Dot-Matrix 형태의 전산신호를 문자나 그림으로부터 추출하는 모듈을 Embedded 시스템으로부터 서버로 이전함으로써 요구되는 하드웨어 성능을 낮추고 사용자에게 필요한 문자열 및 이미지를 자유롭게 사용 가능하도록 하였다.

### 2. 기존 시스템의 구성

기존의 무선 전광판 시스템의 경우 Dot-Matrix 출력을 위하여 출력할 내용에 대한 데이터를 RF통신 등을 이용하여 Embedded 시스템에 받아들인 후 이를 조합형 글자나 내장된 완성형 글자로 재조립 한 후 출력하는 방식을 사용한다. 그러나 조합형 글자의 경우 그 글자를 조합하는데 embedded 코어의 처리량을 많이 사용하게 되는 단점이 있다.

주로 사용하는 Atmel社의 마이크로컨트롤러 코어의 경우 128K 바이트 ISP(In-System Programmable) 플래시 메모리, 4K 바이트 내부 SRAM, 4K 바이트 ISP (In-System Programmable) EEPROM 메모리를 지니고 있어서 완성형 한글의 경우 일반적으로 사용되는 문자의 수가 기본 2384자로 대부분의 메모리를 차지하게 되는 결과를 볼 수 있으며 16\*16 Dot-Matrix 기준의 완성형 폰트는 그림 1과 같다.

```

2256 #include <avr/io.h>
2257 #include <avr/interrupt.h>
2258 #include <avr/eeprom.h>
2259 #include <avr/wdt.h>
2260 #include <avr/delay.h>
2261 #include <avr/pgm_flash.h>
2262 #include <avr/usb.h>
2263 #include <avr/usbdev.h>
2264 #include <avr/usb_lwip.h>
2265 #include <avr/usb_lwip.h>
2266 #include <avr/usb_lwip.h>
2267 #include <avr/usb_lwip.h>
2268 #include <avr/usb_lwip.h>
2269 #include <avr/usb_lwip.h>
2270 #include <avr/usb_lwip.h>
2271 #include <avr/usb_lwip.h>
2272 #include <avr/usb_lwip.h>
2273 #include <avr/usb_lwip.h>
2274 #include <avr/usb_lwip.h>
2275 #include <avr/usb_lwip.h>
2276 #include <avr/usb_lwip.h>
2277 #include <avr/usb_lwip.h>
2278 #include <avr/usb_lwip.h>
2279 #include <avr/usb_lwip.h>
2280 #include <avr/usb_lwip.h>
2281 #include <avr/usb_lwip.h>
2282 #include <avr/usb_lwip.h>
2283 #include <avr/usb_lwip.h>
2284 #include <avr/usb_lwip.h>
2285 #include <avr/usb_lwip.h>
2286 #include <avr/usb_lwip.h>
2287 #include <avr/usb_lwip.h>
    
```

그림 1. 16\*16 Dot-Matrix 기준의 완성형 폰트

기존의 시스템에 맞게 될 경우 Embedded 시스템은 데이터를 받아들이고 그 안에서 다시 조합형 글자를 생성하거나 완성형 폰트를 사용하며 사용할 폰트의 생상과 이동에 대한 설정을 마친 후 다시 출력해야 하는 불편함을 지닌다.

또한 완성형 폰트를 검색하기 위해서는 한 글자를 찾아내기 위하여 2384자의 글자를 저장하여 놓고 있다가 각각의 글자에 맞는 Dot-Matrix 형태의 글자를 찾아야 하며 이는 시스템 제작시 미리 지정된 행렬만큼의 Dot-Matrix 만을 사용 가능하게 한다.

Dot-Matrix의 높이가 16pixel 일 경우 그 높이만큼의 글자를 구성하기 위해서는 정방형의 행렬에 16개의 각 행만큼의 데이터를 필요로 하며 또한 각 글자들에 대한 Font를 다양하게 한다면, 요구되는 데이터의 양은 필요한 Font종류의 곱만큼 늘어나게 된다.

기존 시스템의 예상되는 메모리의 양은 문자열을 비교하기 위한 한글 2384자, 영문 26자, 숫자 10자 조합의 char 형태의 배열 총 2420자와 이를 실제 문자열로 출력하기 위한 int 형태의 각 글자 하나당 16개의 배열이 필요하게 되는데 이를 계산하면 41104\*2byte의 공간을 낭비하는 것을 알 수 있다.

각 폰트를 제작하는데 필요한 프로그램 역시 무료로 공개된 것이 없고, 판매되는 것 역시 고정된 pixel에 대한 Dot-Matrix 배열을 생성하게 되며 이는 개발자에게 많은 시간을 요구되며 기존의 시스템 흐름도는 그림 2와 같다.

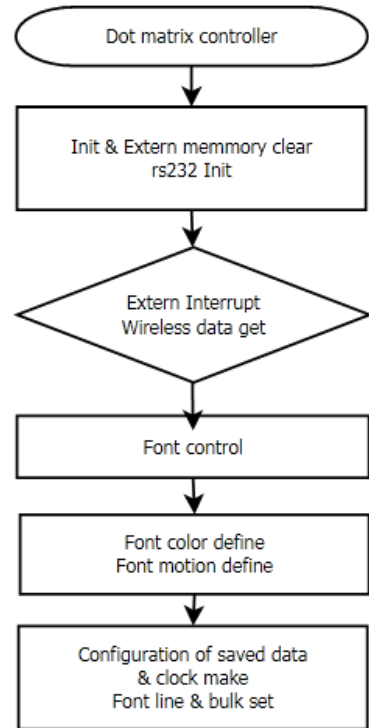


그림 2. 기존의 시스템 흐름도

3. 제안된 시스템의 설계

본 논문이 제안하는 방법은 기존의 부하가 예상되고 시스템의 비용을 높이는 부분을 서버로 이전함으로써 시스템의 필요 성능을 줄이고 또한 WiFi를 하여 네트워크상의 서버와 연동 가능하게 함으로서 AP의 범위 하에서 출력될 전광판의 수와 그 크기를 자유롭게 변동할 수 있도록 하였으며, 제안된 시스템 구성도는 그림 3과 같다.

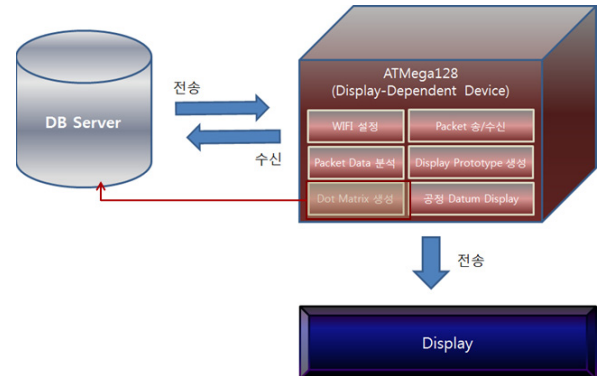


그림 3. System 구성도

기존의 시스템의 예로 들었던 것과 다르게 제안된 설계에서는 ATmega2560을 사용하였는데 이는 대형 전광판이 하나의 출력 Dot-Matrix 가 아닌 다중의 대형화한 Dot-Matrix 모듈을 연결하여 개발하는 경우가 많기 때문이다. 따라서 ATmega2560은 ATmega128에 비하여 충분한 출력 개수를 유지시킬 수 있고 또한 RS232 통신 처리가 가능하여 개발에 사용한 WiFi 모듈을 제어하는데 적합하다.

실제 사용된 모델은 LK-AVR2560D-V01 과 Wiznet wizfi-220 모듈로서 둘 사이에는 RS232 통신을 통한 제어 및 네트워크 송수신이 가능하다. 설계된 하드웨어의 구성은 그림 4와 같다.

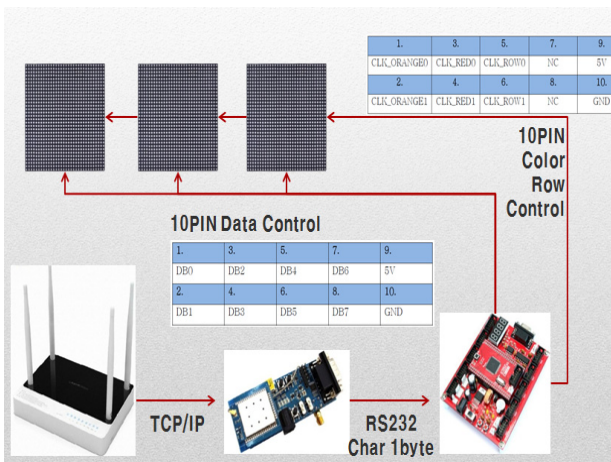


그림 4. 하드웨어 구성도

ATmega2560 모델에 내장한 프로그램은 서버로 문자열에 대한 Dot-Matrix 출력 내용 생성 모듈을 이전하여

단순하고 빠른 처리율을 보장할 수 있도록 설계하였으며 WiFi를 통하여 받아들여진 출력될 Data는 WizFi-220 모델을 통하여 RS232 신호로 변경되어 ATmega2560에 전달된다. ATmega2560의 데이터 수신은 그림 5와 같다.

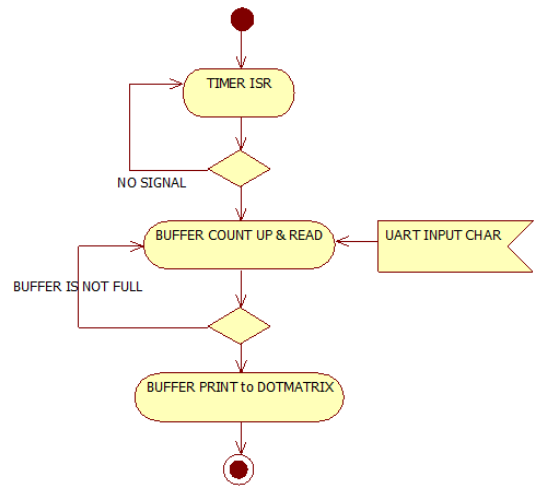


그림 5. ATmega2560 데이터 수신

#### 4. 제안된 시스템의 구현

서버는 Dot-Matrix를 생성하고 WiFi module의 IP를 입력하여 전광판에 출력해야 하는 Data를 지속적으로 전송하는 방식으로 설계되었다. Font의 흐름처리나 형태의 경우에도 서버에서 필요한 스트링에 대한 모형을 찾고 글자 사이의 자간을 구분하며 필요한 문자의 실제 Dot-Matrix 데이터를 찾아서 완성된 형태를 자신이 저장한 다음 필요한 이동거리 만큼 글자를 이동시켜 전송시키는 형태이다. 만들어진 프로그램의 구조는 그림 6과 같다.

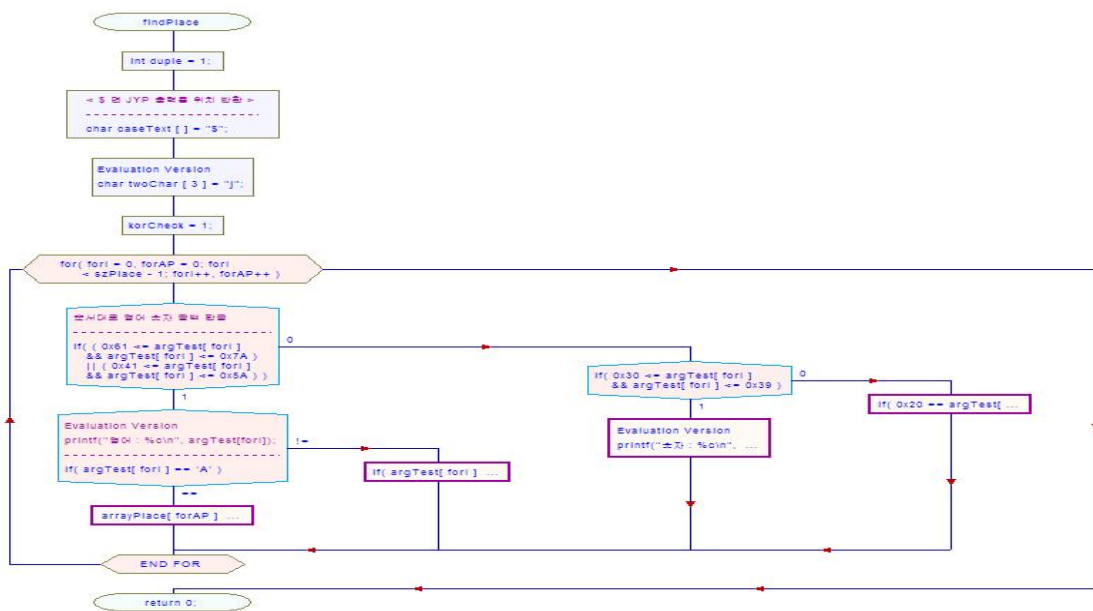


그림 6. 서버 Flowchart

서버에서 전광판 출력에 필요한 Dot-Matrix 값들을 행렬에 저장한 다음 소켓을 활용하여 target 전광판에 전송하게 되는데 16\*16pixel 의 전광판 한 모듈에 필요한 데이터는 32Byte 이다. 한 ATmega2560 보드에서 개발에 사용한 모델의 경우 8개의 전광판 모듈을 개별적으로 제어할 수 있게 구성하였으며 서버에서 전송되는 부분은 그림 7과 같다.

```
//send 32*2bytes
for(int i=charCnt-1; i>charCnt-1-dotSetNum; i--){
    for(int j=0; j<16; j++){
        retval = send(sock, (char *)&dotTemp[i][j], 2, 0);
    }
}
```

그림 7. 서버 내 전송 소스

ATmega2560 에 내장된 Embedded 시스템은 서버에서 그림 7과 같이 전송된 데이터를 중간의 처리과정 없이 바로 출력하게 된다. 단 WiFi 모듈로부터 서버에서 전송된 출력 데이터를 수신하기 위해서는 ATmega2560 과 WiFi 사이의 Baud Rate를 맞추어 줘야 하며 이는 RS232 통신을 활용하기 위해 꼭 필요한 부분이다. 내장 프로그램의 구조는 그림 8과 같다.

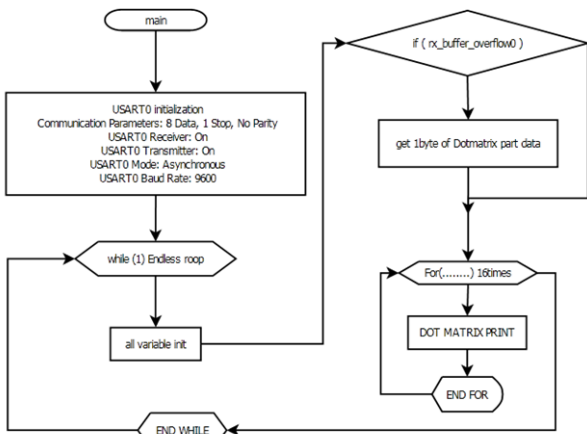


그림 8. 임베디드 시스템 구성도

Embedded 시스템에서 서버로부터 전송된 데이터가 WiFi 모듈로부터 변환되어 RS232 통신을 통하여 들어온 신호는 ATmega2560 의 USART0\_RX\_vect ISR(Interrupt Service Routin) 에 의하여 수신되며 그 과정은 그림 9와 같이 동작한다.

이에 의하여 임시로 저장된 출력 Data는 최종적으로 전광판 모듈의 LED Cathode와 Anode 출력을 통하여 가지적으로 보이게 되며 전광판 모듈 8개는 동시적으로 출력된다.

```
ISR(USART0_RX_vect){ //rx interrupt Service routine
    char status,data=0;
    status=UCSR0A;
    data=UDR0;
    if ((status & (FRAMING_ERROR | PARITY_ERROR | DATA_OVERRUN))==0) //no error?
    {
        rx_buffer0[rx_wr_index0++]=data;
        if (rx_wr_index0 == RX_BUFFER_SIZE0) rx_wr_index0=0;
        if (++rx_counter0 == RX_BUFFER_SIZE0) //rx_buffer full?
        {
            rx_counter0=0;
            rx_buffer_overflow0=1;
        }
    }
}
```

그림 9. ISR

### 5. 결론

본 논문에서는 급속도로 발전된 무선 네트워크 환경을 활용한 전광판 출력 시스템을 제안 하였다. WiFi 환경 하에서 Socket을 통한 네트워크에 접속할 수 있는 모든 전광판은 하나의 서버를 통하여 관리되어 질 수 있으며 그 전광판에 필요한 Embedded 시스템의 비용을 절감할 수 있다.

이 시스템의 장점은 크게 두 가지로 볼 수 있다. 한 가지는 사용자에게 다중 전광판의 출력 내용을 직접 입력할 필요 없이 서버를 통하여 쉽게 관리할 수 있다는 점이다. 다른 하나는 전광판 관리자에게 출력할 폰트나 출력 양식을 서버로 집중 관리하게 함으로서 쉽게 변동할 수 있게 한다는 점이다.

전광판은 LED 산업의 발전과 시장의 확대로 인하여 광고물로서 쉽게 접근할 수 있다는 장점이 있으며, 높은 가시거리로 인하여 대형화와 복잡한 도시나 공장 속에서 사용하기 좋다는 장점이 있다. 따라서 제안하는 시스템은 잠재적으로 상업적인 가치가 높다고 볼 수 있다.

### 참고문헌

- [1] 김범석, 김성곤, 한필희, 박수영, 이의선, 최우영, "Zigbee를 이용한 무선 실시간 전광판", 산업기술연구센터 논문집 제 28권 pp.26-30, 2009.
- [2] 이연석, 윤영준, "문자기반 모바일 네트워크를 이용한 임베디드 전광판의 원격제어 시스템의 구현", 제어로봇시스템학회 논문지, 제 12권 1호, pp.72-77, 2006.
- [3] 권철우, 최성일, 정구일, 황희용, "RF통신을 이용한 전광판 시스템의 구현", 반도체장비학술심포지움, 제 2001권, pp.115-118, 2001.
- [4] 최종호, "저가형 다기능 LED 전광판 시스템" 산학기술연구소논문집, 제 7호, pp.171-189, 1999.
- [5] 김명희, 김정만, 연학현, "ATmega128을 이용한 LED 신호등 제작", 정보통신연구소 연구논문집 제7권 1호, pp.41-44, 2011.
- [6] 윤덕용, AVR ATmega128 정복, pp. 212-221, Ohm출판사, 2006.
- [7] 한새론, 김성환, 최관순, 임종식, 김동식, 이순흠, 전홍구, "ATmega128을 이용한 원격 전기회로 실험실의 구현", 한국정보기술학회논문지 제8권 11호, pp.215-224, 2010.