한국산학기술학회논문지
Vol. 11, No. 3, pp. 951-957, 2010

GSM 핸드폰을 위한 폰트 처리 시스템의 설계 및 구현

이상범[1], 이용훈[1*]
[1]단국대학교 전자계산학과

Developement of a Font Processing System for GSM Mobile Phone

Sang Bum Lee[1] and YongHun Lee[1*]
[1]Dept. of Computer science, Dankook University

요 약 본 논문에서는 GSM 핸드폰 단말기에 사용할 폰트를 제작 처리할 수 있는 효율적인 시스템을 제안한다. 문자를 컴퓨터상에 표현하는 방법으로는 초기에는 아스키코드(ASCII code)가 개발 되어 사용하다가 문자 표현의 개수의 한계로 유니코드(Unicode)가 개발 되었으며 이에 대한 표준화 작업이 계속 되고 있다. 유니코드를 무선 단말기 상에 적용하고자 하는 노력이 계속 되어 왔으나 각 나라마다 각기 다른 언어를 표현해야 하는 무선 단말기 상의 하드웨어적 제한이 있기에 이를 개선하기 위한 많은 시간과 노력이 필요하게 되었다. 본 연구에서는 이러한 문제를 개선할 수 있는 방법으로 유니코드를 무선 단말기에 적용시키는 일련의 과정을 단축하여 처리할 수 있는 효율적인 폰트 처리시스템을 설계 및 구현하였다.

Abstract In this thesis, we propose a font development system that can handle various fonts efficiently in the GSM mobile terminals. The ASCII code was widely used to express characters on the computer in the beginning but it has limitation for representing many characters. Recently, Unicode was developed to add more characters. Researches on code systems are still on going to express the characters more efficiently. Attempt of applying this kind of Unicode to the mobile terminal didn't work efficiently since there are too many characters for various languages. In this research, we designed and developed a font system to shorten processing time and efforts that apply Unicode to mobile terminals to solve these problems. Our system can save processing time and efforts since it reduces the meaningless processing compared to other systems.

Key Words : GSM, Mobile, Font, Unicode, ACSII

1. 서론

90년 후반부터 무선 단말기 보급이 늘어나기 시작하여 지금은 대부분 국민들이 핸드폰을 소유하고 있다. 무선단말기 제조업체는 기술 개발에 많은 투자를 하여 우수한 단말기를 개발하여 국내 소비자를 만족시킬 뿐만 아니라 해외에서도 그 점유율을 높이고 있다. 해외에 수출되는 무선 단말기는 그 나라에 정서에 맞은 인터페이스와 기호에 맞는 기능을 갖추어야할 뿐만 아니라 해당 국가의 언어를 반드시 지원해줘야 한다. 단순히 언어를 지원해주는 것으로 그치지 않고 그 언어를 표현하는 다양한 폰트도 제공된다면 소비자 또한 좋은 선호도를 보일 것이다.

무선 단말기의 액정 모니터는 상대적으로 작지만 다양한 문자를 보여줘야 하기 때문에 문자를 나타내는 폰트 처리에 있어서 상당한 작업과 노력이 요구된다. 예를 들

이 연구는 2009년도 단국대학교 대학연구비의 지원으로 연구되었음.
*교신저자 : 이용훈(karismanj@dankook.ac.kr)
접수일 09년 10월 17일 수정일 (1차 09년 12월 21일, 2차 10년 02월 03일) 게재확정일 10년 03월 18일

면, 중국어인 경우 무선 단말기 상에 적용되는 한자의 글자 수는 대략 3만자 정도가 필요하여 이들 각 글자의 폰트 제작이나 변경을 고려한다면 개발 시간과 비용은 엄청나다고 할 수 있다. 따라서 단말기를 제조하여 수출하는 회사가 각 국가별 언어를 지원하는 폰트 제작에 너무나 많은 시간을 소모할 수밖에 없어 폰트 처리만을 전적으로 담당하는 업체에 의뢰하는 경우가 많다.

본 논문에서는 이러한 기존의 문제점을 개선하여 불필요한 작업을 없애고 폰트 처리의 능률과 효율을 극대화 시키기 위한 폰트 제작 시스템을 개발하였다. 폰트는 글자의 형태와 크기로 표현되는데, 지금까지는 같은 글자에 다양한 형태와 크기를 별도의 비트맵으로 수작업을 처리를 해왔다. 특히 무선 단말기는 성능과 화면의 특성이 다르기 때문에 같은 글자라도 단말기별로 세심한 폰트처리 작업을 수행해야하는 경우가 많다. 제안된 시스템은 표준화된 하나의 글자를 비트맵으로 제작하고 이를 코드화로 변환시켜 저장하며 프로그램으로 크기를 쉽게 바꿀 수가 있을 뿐만 모양도 쉽게 수정할 수 있도록 도와준다. 따라서 이 시스템의 가장 큰 장점은 국가별의 언어를 비트맵으로 제작하여 한꺼번에 처리 할 수 있어 폰트 개발 비용과 시간을 많이 줄일 수가 있게 되었다. 특히 이러한 시스템은 해외의 다양한 문자를 지원해야하는 GSM 무선 단말기의 폰트 처리를 위해 더욱 더 필요하다고 할 수가 있다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구인 아스키 코드와 유니코드 부분에 대한 발전 과정과 필요성을 살펴보고, 3장에서는 GSM 핸드폰의 소프트웨어 구조에 대하여 다루어 보았다. 4장에서는 폰트 처리 시스템에 대한 설계와 구현된 내용을 소개하고 있으며, 마지막으로 5장에서는 향후 과제 및 결론을 기술하였다.

2. 관련 연구

2.1 문자 코드 시스템

아스키 코드(ASCII : American Standard Code for Information Interchange)는 미국 규격 협회인 ANSI (American National Standards Institute) 에 의해 개발되었으며 컴퓨터나 인터넷상에서 텍스트 파일을 위한 가장 일반적인 코드 표현 양식이다. 아스키 코드는 알파벳 문자나 숫자 그리고 특수 문자들이 7비트로 표현되며, 총 128(=27)개의 문자를 정의할 수 있다. 유닉스나 도스 기반의 운영체계(윈도우 NT는 제외)는 텍스트 파일을 표현하는데 아스키 코드를 많이 사용하였다. 7비트로 표현되는 문자의 개수는 컴퓨터가 영어만을 사용한다면 가능하겠지만 컴퓨터는 모든 나라의 언어를 표현하기에는 역부족이다. 따라서 새로운 코드 시스템이 필요하게 되었다[1-4].

유니코드(Unicode)는 세계 각국의 언어를 통일된 방법으로 표현 할 수 있게 제안된 국제적인 코드 규약으로 8비트의 아스키코드를 16비트로 확장한 국제 표준 코드이다. 8비트로 표현 할 수 있는 256자는 영어나 라틴계 등에서는 문제가 없으나, 한국, 일본, 중국, 아랍 등의 다양한 문자들을 표현하는 데는 한계가 있다. 또한 각 나라마다 같은 코드 값에 다른 글자를 쓰게 된다면 국제간의 원활한 자료 교환이 불가능하기 때문에 코드를 16비트 체제로 확장해서 65,536자의 영역 안에 전 세계의 모든 글자를 표시하는 표준안이다.유니코드가 다양한 문자 집합들을 통합하는 데 성공하면서 컴퓨터 소프트웨어의 국제화와 지역화에 널리 사용되게 되었으며, 비교적 최근의 기술인 XML, 자바, 그리고 최신 운영 체제 등에서도 지원하고 있다[1-4].

기존 문자 세트 표준 (KSC 5601 혹은 일본의 JIS X 0208 등)이 사용된다고 할 경우 다양한 문자 세트 표준사이의 변환을 위해서는 N * (N-1)개의 변환 테이블이 필요하다. 이는 구현상 상당한 공간을 낭비하게 된다. 따라서 기존 문자 세트 표준 사이의 변환을 할 때, 일단 유니코드로 변환한 후 다른 문자 세트 표준으로 변환 하게 되면 변환 테이블이 2 * N개만 필요하게 된다. 이로써 프로그램 및 시스템이 한 번에 기존의 다양한 문자 세트를 다룰 수 있으며 문자 세트 변환을 다루는 프로그래밍이 상당히 간편해 진다. 유니코드가 문자 세트 표준이 되기 위해서는 상당부분의 기존 문자 세트 (ISO-8859-1, KSC-5601 등)의 문자를 모두 포함하며 기존 문자 세트의 문자를 유니코드로 변환한 후 다시 원래의 문자 세트로 변환하여도 원래의 자료가 변경되지 않고 그대로 복구 되어야 한다. 한글은 AC00~D7A3 코드 자리에 위치하고 있다. 이 표준에는 ISO 10646 문자 집합, 문자 인코딩, 문자 정보 데이터베이스, 문자들을 다루기 위한 알고리즘 등을 포함하고 있다[3,4].

2.1 초기 준비 단계

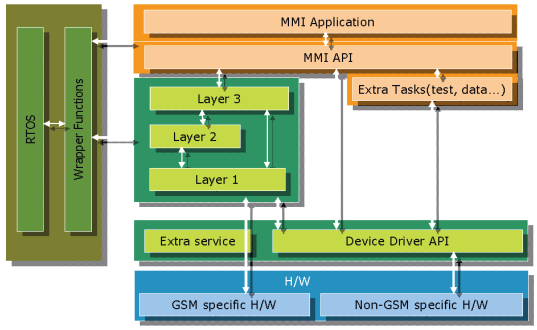
폰트는 문자 정보를 디지털형태로 표현하는 매체로 컴퓨터에 의해 타입과 스타일 그리고 크기가 생성될 수 있도록 개발되었다. 이를 개발하는 방법은 일정크기의 매트릭스에 활자모양을 수작업으로 점을 찍어 처리하도록 하는 비트맵 방식 폰트와 글자의 외곽선을 수식으로 정의하여 처리하는 벡터방식의 폰트로 나눌 수 있다 [5,6]. 폰트처리 시스템은 기업체에서 한글편집기나 프린터에 적용되는 것으로 개발했기 때문에 주로 특허로 출현된 것이 많다. 비트맵 방식인 경우 일반적인 폰트 시스템은 각

글자에 해당하는 폰트를 별도의 이미지로 만들어 사용하기 때문에 하나의 글자라도 크기, 모양, 색상 등에 따라 많은 이미지 정보가 필요하다. 임베디드 시스템을 위한 폰트 시스템[7], 그리고 다 목적 폰트 시스템을 위한 연구[8]가 제안되었으나 본 연구와는 그 방향이 많이 달라서 비교 분석은 하지 않았다. 그 외 폰트관련 연구로는 트루타입폰트로부터 한자의 외곽선 정보를 자동적으로 추출하여 벡터 형식으로 전환할 수 있는 연구가 있다[9]. 한편 COVIS(COmics Rendering system on VIdeo Stream)은 만화의 디지털 작업을 위해 이를 지원하는 스타일 폰트를 설계하여 제작하였다 [10]. 그 외 폰트의 내용을 인식하는 연구가 진행되고 있다[11,12].

3. GSM 단말기의 소프트웨어 구조

GSM(Global System for Mobile Communications, Groupe Spécial Mobile)은 전세계에서 가장 널리 사용되는 개인 휴대 통신 시스템으로 TDMA (Time Division Multiple Access) 기반의 통신 기술이다. GSM은 1982년에 규격제정이 착수된 이래 10년 동안의 기술 개발과 규격 보안을 통하여 1992년에 처음으로 GSM 상용 서비스가 시작되었다. 한국에서는 개인 이동 통신 시스템으로 CDMA(Code Division Multiple Access)와 WCDMA (Widedband Code Division Multiple Access)를 사용하고 있어, GSM 방식의 휴대전화를 사용하고 있지는 않지만, 많은 이동 통신 제조업체들이 수출용으로 GSM 방식의 단말기를 개발하고 있다. 최근 GSM 기술의 진화에 따라 보다 발전된 형태의 GSM 기술 표준이 발표되었다. 1997년 배포된 무선 데이터 통신 규격 GPRS(General Packet Radio Service), 1999년 배포된 EDGE(Enhanced Data Rates for GSM Evolution) 등이 그것이다[13,14].

GSM방식의 핸드폰의 소프트웨어 아키텍쳐는 아래의 그림 1과 같다. RTOS는 태스크 스캐쥴링 선점을 위한 멀티태스킹 기능, Time-Sliced 스캐쥴링, 정적/동적 태스크의 생성, 메시지큐에 대한 태스크간의 통신, 메모리/자원 관리 그리고 모든 프로세서에 기본적인 인터페이스를 제공한다. 특히 MMI는 단말기의 각각의 부분을 컨트롤 하도록 정의한다. 주요 기능으로는 사용자에게 하드웨어의 인터페이스를 제공하는데, 주요한 것으로는 키보드, 디스플레이, LED, 백라이트 그리고 모든 오디오와 비주얼 하드웨어 장치가 포함되어 있다. 따라서 키보드 입력을 허용하고 디스플레이 스크린 텍스트와 그래픽스를 표시할 수 있으며 오디오를 들을 수 있도록 환경을 제공해 주고 있다[14-16].

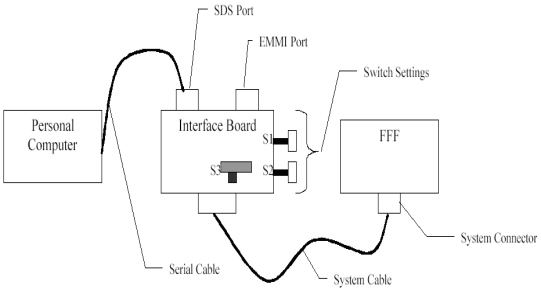


[그림 1] GSM 핸드폰 소프트웨어 아키텍처

MMI API의 사용은 GSM 시스템 소프트웨어 개발을 위한 소프트웨어 구조에서 가장 중요한 개념이다. API는 소프트웨어 개발의 간결성과 숨김(isolation) 그리고 추상화(abstraction)를 제공한다. 특히 MMI API는 다른 소프트웨어의 표준 인터페이스를 제공하며 이는 3개의 레이어로 구성되어져 있는데, 레이어1은 상위 계층과의 인터페이스 역할을 하며, 모든 하드웨어의 스펙 소프트웨어를 포함 하고 있다. 레이어2는 레이어1과 레이어3의 Data link를 수행한다. 또한 레이어3은 모든 동작 상태와 시나리오를 위한 과정과 시퀀스를 포함한 protocol stack의 중요한 부분이다[17,18].

본 논문에서 소개 하고자 하는 시스템은 수출용 GSM 단말기를 위한 폰트 처리 시스템이다. 따라서 이 시스템은 그림 1과 같은 GSM 단말기의 설계 구조에 따라 최종 사용자가 접하게 되는 최종적인 인터페이스 부분인 MMI 어플리케이션 상에 데이터를 적용하여 처리하게 된다.

일반적으로 핸드폰관련 시스템 소프트웨어는 PC에서 개발하여 실행파일로 만든 후 이를 다운로더 시스템을 이용하여 핸드폰 단말기에 로딩 시킨다. 그림 2는 PC와 핸드폰을 연결시키는 다운로더 시스템의 구성도를 보여주고 있다[18].



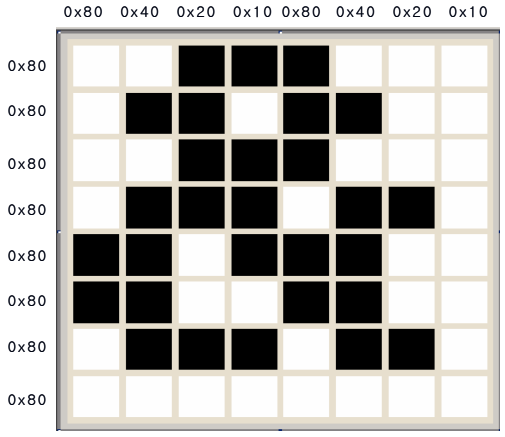
[그림 2] 핸드폰 다운 로더 시스템

4. 핸드폰 폰트 처리 시스템의 구현

4.1 폰트 시스템의 설계

폰트(Font)는 글자의 형태와 크기를 가리키는 용어로 사용되었으나 컴퓨터 환경에서는 글자의 형태와 색을 지칭하는 서체와 같은 의미로 사용한다. 영문의 경우에는 고정간격(Fixed) 폰트와 비례간격(Proportional) 폰트를 사용하며 대부분의 폰트는 비례 간격이다[19,20].

폰트는 크게 비트맵 폰트와 벡터 폰트로 나눈다. 비트맵 폰트의 경우에는 여러 개의 점으로 하나의 글자 패턴을 기록하는 방법이며 크기에 따라서 전체적인 메모리 용량이 늘어날 수 있다. 지정된 크기가 아닌 글자를 표시할 경우 엘리어싱 현상이 발생 할 수 있다. 벡터 폰트는 트루타입 폰트라고도 하며 글자를 자유롭게 확대, 축소, 변형시켜도 글꼴의 찌그러짐이 발생하지 않는 폰트를 말한다[5,21,22]. 글자 모양을 나타내기 위해 윤곽선을 수학적인 함수로 표현 한다[7]. 프린터의 해상도는 dpi로 표현하며 모니터 스크린의 해상도는 픽셀로 표현하기 때문에 출력물에서는 벡터 폰트가 좋다. 일반적으로 무선 단말기는 비트맵방식의 폰트를 사용하고 있다.

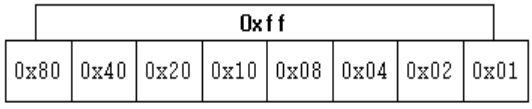


[그림 3] 폰트 데이터 입력 형태

제안된 폰트 시스템은 위 그림 3와 같이 한 글자 (예를 들어 '&')를 비트맵 형태로 먼저 수작업을 통해 개발한다. 즉, 사각형 모양의 픽셀로 이루어진 비트맵 위에 글자 모양대로 픽셀들을 칠하는 방식으로 폰트를 만드는 것을 말한다. 그리고 각 픽셀은 위치 정보가 있으므로 이러한 정보를 해당 글자의 데이터로 전환을 하게 되는데, 검은색으로 칠한 것은 비트의 값을 1로 만드는 것을 의미한다. 해당 데이터를 적용하는 실제 데이터는 아래코드와 같은데, 예를 들면 첫 번째 값 "0x38"은 위 행렬의 첫 번째 열을 비트 값으로 표현하면 "00111000"이 되는데 이를 16진수 값으로 전환한 것이다. 이러한 방법을 인용하여 폰트 시스템에 문자를 입력하는 것과 동시에 이를 코드화 할 수 있게 된다.

```
#define FONT_8 x 8 Color
{ 0x38, 0x6c, 0x38, 0x76, 0xdc, 0xcc, 0x76, 0x00 }
```

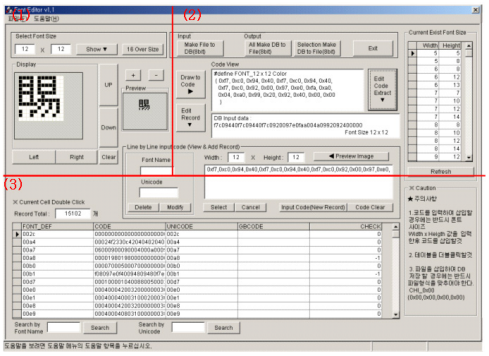
아래의 그림 4은 비트맵 폰트에 찍히는 데이터 값 한 줄을 표현한 것이다. 그림에 보이는 데이터 값을 적용하면 해당 부분에 폰트가 찍히게 된다. 두 점이나 그이상의 부분에 폰트를 찍으려 한다면 해당 데이터 값을 더해주기만 하면 된다. 예를 들어 만약 8비트 모두 폰트를 찍는다고 한다면 데이터 값은 0xff 가 되는 것과 앞의 네 칸을 찍는다 하면 0xf0 가 되는 것이고 각각의 한 칸씩을 찍는 다면 아래 칸칸 마다 적혀있는 값을 적용 하면 되는 것이다.



[그림 4] 8비트 데이터의 구성

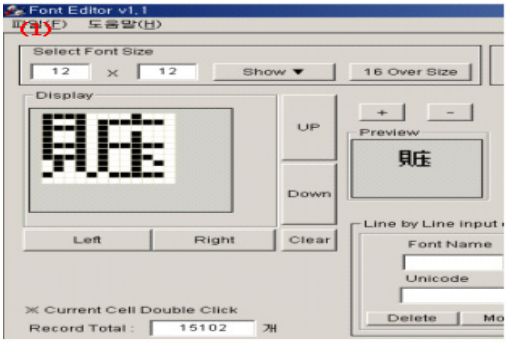
4.2 폰트 처리 시스템의 개발

본 절에서는 위에서 기술한 폰트 처리 중 비트맵 처리에 대한 코드 자동 생성 과정을 좀 더 자세히 기술하고자 한다. 제안된 폰트 처리 시스템의 주안점은 수많은 비트맵 폰트를 어떻게 효율적으로 빠른 시간 안에 처리를 하는가에 대한 해결방안을 제시한 시스템이다. 시스템은 Visual C++를 사용하여 구현하였으며 윈도우즈 기반의 운영체제에서 실행된다. 시스템의 전체적인 인터페이스는 그림 5과 같으며 좌측 상단부분(1)은 폰트 입력 및 제작을 위한 부분이며 우측 상단부분(2)은 데이터(해당 Character 에 해당하는 코드) 출력을 나타내고 그림의 하단부분(3)은 데이터 저장 (실제 데이터베이스에 저장되어 있는 데이터) 부분을 표현한다. 글자 입력 또는 데이터베이스로부터 읽어온 폰트를 코드 형식으로 변환해주는 프로그램이다. 변환 과정 중간에 미리 보기 창에서 화면에 픽셀 단위로 된 부분에 점을 찍어서 폰트의 모양의 변경할 수 있으며 또한 빈 화면에 픽셀을 찍어서 새로운 폰트를 만들 수도 있다.



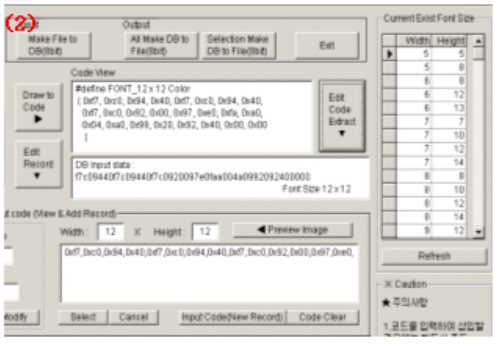
[그림 5] 폰트 시스템의 구성 화면

그림 6는 폰트를 입력하는 부분으로서 화면에는 12 x 12기준으로 한자 한 글자를 입력한 모습이다. 글자입력은 마우스로 클릭 및 드레그하여 문자를 입력하도록 구성하였는데 이는 폰트가 찍히는 12 x 12화면이 비트맵으로 인식되기 때문이다. 오른쪽에는 미리보기 창을 만들어 실제로 입력하는 것과 보여지는 것에 대한 차이를 미연에 방지하도록 하였다.



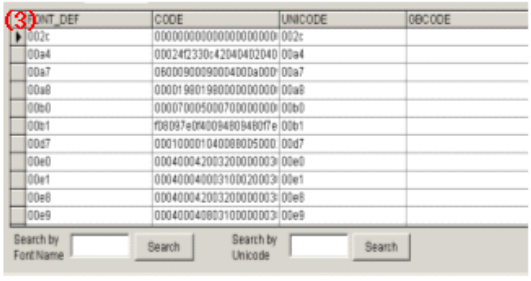
[그림 6] 폰트 처리 시스템

그림 7는 위 그림 6에서 입력한 글자를 16진수 데이터 값으로 출력한 화면으로 해당 폰트의 코드 값을 나타낸다. 이렇게 원하는 모양의 문자를 직접 무선 단말기에 적용하는 코드 값으로 변환하여 핸드폰단말기에 적용할 수 있도록 폰트 데이터의 코드 값을 출력하도록 하였다. 출력형태는 실제로 무선 단말기에 입력되는 #define 선언문까지 출력 되도록 한 것과 순수 데이터 값만 출력되도록 한 것 두 가지의 형태가 있다.



[그림 7] 데이터 출력

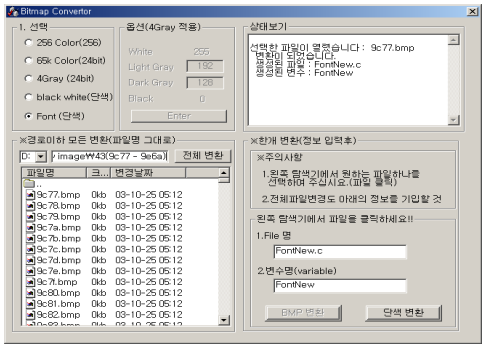
위의 그림 7의 오른쪽 부분은 이미 제작하여 데이터베이스에 저장되어있는 폰트들의 정보를 볼 수 있게 하였다. 사전에 제작되었던 폰트들이 어떤 형태와 사이즈들로 저장이 되어있는 지를 나타내고 있다. 이를 확대하여 보여주는 것이 그림 8이며 데이터의 형태와 유니코드 값을 비롯한 저장되어 있는 정보를 보여주도록 한 것이다. 데이터베이스는 MS Office에서 기본적으로 제공 되는 Access를 사용하였고 데이터베이스와의 연결은 ODBC를 사용하여 연결하였다. 데이터베이스는 언제라도 다른 형태의 데이터베이스로 바꿀 수 있도록 유동적으로 구성이 되어 있다. 대용량의 데이터 처리를 원할 경우 데이터베이스의 테이블 값과 필드 값만 추가를 한다면 대용량 처리 데이터베이스의 변경이 용이하다. 유니코드는 ISO/IEC 10646-1에 의거 하여 자동적으로 해당 코드가 저장되도록 구성 하였고 유니코드 표현 방법으로는 UCS (Universal Multiple-Octet Coded Character Set) 을 이용하였다[19.20].



[그림 8] 데이터 저장부분

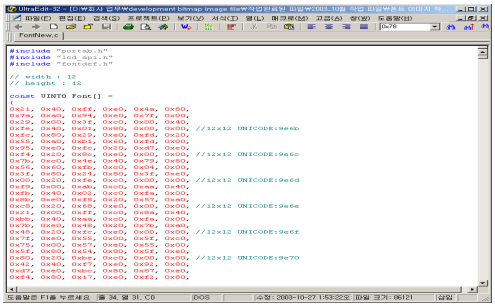
아래의 그림 9는 BMP 이미지 파일로 만들어진 폰트를 읽어 들여 16진수 데이터 값으로 변환 해주는 역할을 수행한다. 또한 256 및 65k color bit map, 4Gray, black & white(단색), 폰트 등의 비트맵을 처리할 수 있도록 구성하였으며 처리량은 컴퓨터의 사양에 따라 달라지겠지

만 대략 P-4 1.8GHz의 사양을 가진 컴퓨터에서 30000 Character를 처리하는데 1분이 채 걸리지 않았다. 출력 형태는 그림 10과 같이 하나의 C파일의 형태로 출력 되도록 하였으며 각각의 데이터는 정렬되어 파일내부에 저장된다.



[그림 9] 이미지 처리 시스템

아래의 그림 10과 같이 출력되도록 한 것은 무선 단말기 상에서의 데이터 저장 형태가 그림 10의 출력 형태를 갖기 때문이다. 출력된 결과를 최종적으로 무선 단말기에 적용하면 된다. 이 시스템을 통하여 폰트 이미지 파일로부터 무선 단말기에 사용될 데이터를 추출해 내는 과정을 프로그램화하여 폰트처리에 드는 비용과 시간을 단축시키는 효과를 가져오게 된다.



[그림 10] 데이터 출력 형태

5. 향후 개발 방향 및 결론

GSM 무선 단말기 제조회사는 각 나라의 다른 언어들을 지원하기 위해 폰트를 제작 개발하는데 많은 시간과 비용이 들어간다. 그 이유는 같은 글자라도 다양한 글꼴과 크기에 따라 별도로 비트맵 방식으로 개발되어야 하는데, 무선 단말기는 그 종류가 다양하기 때문에 같은 폰트가 다른 단말기에 제대로 나타나지 않는 경우가 많은데, 이 경우엔 재수정 작업을 해야 하기 때문에 소모되는 비용이 엄청나다고 할 수 있다. 본 연구에선 제안한 폰트 시스템은 하나의 폰트를 개발한 후 이를 코드화 시켜 시스템 저장시켜 활용하는데, 크기는 자동적으로 조정이 가능하며 각 단말기별로 수정이 필요한 경우 기존 폰트를 불러들려 간단한 작업을 통해 수정이 용이하다. 특히 단말기 따라선 폰트를 16비트로 또는 8비트로 전환시켜야 하는데, 이러한 작업도 쉽게 해결된다. 특히 유니코드에 정의된 문자 중, 한 국가에 대한 무선 단말기에 문자 혹은 기호 등을 모두 적용하기란 힘들고 또한 초기에 적용단계에서 많은 시간과 노력을 필요로 한다. 따라서 개발도중 혹은 테스트 과정에서 단말기에 적용 되어있지 않거나 쉽게 표현되지 않은 문자들에 대한 문제점을 해결하고 또한 초기 적용단계에서의 폰트 적용 시간을 단축시키는데 역점을 두고 구현을 하였다. 따라서 이러한 시스템을 개발함으로써 많은 시간과 비용을 단축시키는데 일조를 한 것은 사실이나 여전히 비트맵 파일을 만드는 과정에 상당한 비용과 인력을 필요로 하고 있다. 향후 개발 방향은 글자의 서체를 자동적으로 전환시킬 수 있는 시스템으로 발전시키고자 한다.

참고문헌

[1] http://www.w3c.or.kr/i18n/

[2] 정주원, “ISO/IEC-10646 Universal Multiple-Octet Coded Character Set”,1995.

[3] ISO/IEC 10646-1 "Information technology - Universal Multiple - Octet Coded Character Set (UCS) - ", First edition pp.1-15, May. 1993.

[4] http://www.unicode.org/unicode/standard/principles.html

[5] 김은희, 정근호, 최재영, “트루타입의 합성 글립을 이용한 한글 폰트의 중복성 최소화 방법”, 정보과학회 논문지(B), 제26권, 제10호, 1999.

[6] Sang-Young Mah, "Font Operation Methods in the Mobile Imagies", Foreign Languages Education, Vol. 10, No.4, 2003.

[7] 채영훈, 문승진, “임베디드 시스템을 위한 한글, 일본어, 한자 폰트 제작 및 출력에 관한 연구”, 한국인터넷정보학회 학술발표대회 논문집, 제7권, 제1호, 2006.

[8] 김차종, 강한종, 이수연, “다목적 폰트 제작 시스템”, 한국정보과학회 논문지, 제16권, 제5호, 1989

[9] 장현규, 구상옥, 정순기, “트루타입폰트 기반 자동 획 분할”, 한국정보과학회 32회 추계학술발표회 논문집

Vol.32. No. 2(1) pp. 703-705, 2001.
[10] 이정현 외 3명, “흑백 만화 제작을 위한 스타일 폰트 설계 시스템”, 정보처리학회논문지A 제15권 제2호, pp.75-80, 2008.
[11] 진성아, 주문원, “폰트 밀도함수를 이용한 폰트 타입의 인식”, 한국멀티미디어학회 춘계학술발표논문집, pp.189-191, 2001.
[12] A. Zramdini, R.Ingold, "Optical Font Recognition Using Typographical Features", IEEE Trans. on Pattern Analysis and Machine Intelligence. Vol. 20. No.8. pp.877-882, Aug. `1999.
[13] Siegmund M. Redl, Matthias K. Weber, Malcolm W. Oliphant, "An Introduction To GSM" . Artech House, Inc., 1995.
[14] "GSM Technical Specification" European Telecommunications Standards Institute, 1996.
[15] "Layout Engine" The Mobile Suite Reference library A Reference Guide Version 1.3.1, AU-SYSTEM AB, 2002.
[16] "Mobile Internet Client", The Mobile Suite Reference library A Reference Manual Version 1.1.12, AU-SYSTEM AB. 2003.
[17] "GSM/GPRS Message Sequence Charts", Skyworks Solutions, Inc. Proprietary and Confidential. February 18. 2002.
[18] "GSM/GPRS Man-Machine Interface Application Programming Interface", Software Programmer & Reference manual, Skyworks Solutions, Inc. Proprietary and Confidential. August 1, 2002.
[19] 김경석, “컴퓨터 속의 한글이야기”, 영진출판사, 1995.
[20] 백성수, "C로 구현한 OUTLINE FONT EDITOR", 높이 깊이 출판사, 1994.
[21] 한주현, 정근호, 최재영, “트루타입 폰트에 내장된 한글 비트맵 데이터의 압축”, 정보과학회논문지(B), 제33권, 제6호, 2006.

이 상 범(Sang-Bum Lee) **[정회원]**

- 1989년 12월 : 루우지애나주립대 (전산학석사)
- 1992년 8월 : 루우지애나주립대 (전산학박사)
- 1992년 9월 ~ 1993년 10월 : 전자통신연구원 선임연구원
- 1993년 10월 ~ 현재 : 단국대학교 교수

<관심분야>
소프트웨어공학, 정보검색, 모바일컴퓨팅

이 용 훈(Yong-hoon Lee) **[준회원]**

- 2009년 2월 : 단국대학교 컴퓨터과학과 (공학사)
- 2009년 3월 ~ 현재 : 단국대학교 전자계산학과 석사과정

<관심분야>
정보 검색, 데이터 마이닝, 기계학습