

임베디드 보드를 이용한 미아 위치 정보 시스템

Place Information System for Missing Children using Embedded Board

이 현 태* 이 구 연**
Lee, Hyeon-Tae Lee, Goo-Yeon

Abstract

In this paper, we implement a place information system for missing children using embedded board. In the implementation, the embedded board receives location information from GPS satellites, parsing GGA information which includes current time and location coordinates from the satellite information and sends them to the location management server. The server locates a missing child using the location information from embedded board together with server database. We implement an application program for the system. The program shows current location of a child and its moving path and browses the corresponding map from coordinates mapping.

키워드 : GPS, 소켓, 임베디드보드, 데이터베이스, 자바
Keywords : GPS, Socket, Embedded Board, Database, Java

1. 서론

현대사회로 접어들면서 아동에 대한 관심이 높아지고 사회안전망이 형성되고 있다고 하지만 아이들은 우리가 잠시 방심하고 있는 사이, 예상치도 못한 짧은 시간에도 없어지곤 한다. 경찰청 통계에 따르면 매년 2,500~3,000명의 아동이 미아로 발생하고 있고, 그 중 96% 이상의 아동이 1일~2일 사이에 가정으로 돌아가고 있다. 대부분의 미아는 단기간 내에 안전하게 부모님께 돌아가지만, 장기간 아이를 찾지 못하는 경우의 가정은 심리적, 경제적 고통을 겪게 되며 심한 경우 가정파괴에까지 이르게 된다. 이는 단순한 개인적인 문제가 아닌 사회적인 문제로 대두되고 있다.

이러한 문제점을 해결하기 위하여 미아를 찾기 위한 여러 시스템들이 개발 운영되고 있으나 아직 현실적으로 효율적으로 운영되고 있지 못하고 있는 상황이다. 여러 휴대폰 회사에서는 소지하고 있는 휴대폰을 이용하여 아동등의 위치를 추적할 수

있는 서비스를 제공하고 있으나, 기지국을 중심으로 위치를 찾기 때문에 위치의 정확성을 제공하는 데 어려움이 있다. 한편 한국위치정보라는 회사는 자체 기지국을 세워 정확한 위치정보를 제공하고 자 하였으나, 서비스의 활성화가 이루어지지 않아, 현재는 서비스를 제공하지 않아, 일반인들은 서비스를 받고 있지 못하고 있는 상황이다.

우리는 이러한 문제점을 해결하기 위하여 해당 관리자에게 개인 위치정보 서비스를 제공하여 미아, 배회 등으로 인해 지속적인 보호가 필요한 어린이, 치매 노인 등 위급상황발생 가능성이 높은 사람의 보호 및 감시를 가능하게 하는 시스템을 구현한다.

우리가 개발한 Place Information System은 현재 여러 서비스로 제공되고 있는 실시간 위치정보 시스템을 도입하여 각 미아보호소에서 서버와 연결된 Application을 통하여 해당 어린이를 관리하는 시스템이다. 본 과제에서는 지역을 놀이공원 같은 곳으로 한정하여 지역 내 감시를 원하는 어린이 및 장애아에게 GPS가 장착된 목걸이나 팔지를 나누어 주어 일정시간마다 대상의 위치정보, 즉 현재 위치와 시간정보를 미아보호소 내에 있는 서버로 전송하게 하여, 해당 대상의 현재 위치 및 이동

* 강원대학교 컴퓨터공학과 학사과정
** 강원대학교 컴퓨터학부 교수, 공학박사,
교신 저자

궤적을 추적관리 할 수 있는 시스템을 구현한다. 이 데이터를 통해 미아 발생 시 미아보호소 내에 있는 서버로 미아의 위치정보를 확인하여 미아발생시 신속하게 미아를 찾을 수 있게 된다.

2. 관련 연구

2.1 GPS

2.1.1 GPS의 정의

GPS는 Global Positioning System의 약자로 위성항법장치를 말하는데 GPS 위성들이 지상의 장치로 신호를 전송한다. GPS 수신기는 신호를 받기만 하는데 지구 궤도에 있는 위성의 신호를 받기 위해서 하늘이 보이는 야외에서만 사용이 가능하며 밀림지역이나 빌딩 등이 많은 곳에서는 정확하게 동작하지 않는 경우도 많다. GPS의 동작은 매우 정확한 시간 제어를 필요로 하는데 각 위성에 원자시계가 장착되어 있다.

각 GPS 위성은 자기의 위치와 현재시간 정보를 전송한다. 모든 위성들은 동기화되어 동작을 함으로써 정확히 동시에 이 데이터 전송작업을 계속적으로 반복할 수 있다. 이 신호는 빛의 속도로 이동하여 GPS 수신기에 전달되는데 위성의 위치가 서로 다르기 때문에 각 신호가 도착하는 시간은 아주 조금씩 차이가 난다. 위성과의 거리는 수신기에 도착하는 신호의 시간을 계산함으로써 결정된다. 수신기가 최소한 4개 위성의 거리를 계산하게 되면 자기의 위치를 3차원 적으로 산출할 수 있다 [1].

2.1.2 GPS의 원리

GPS수신기는 위성의 송신정보를 통해 위성의 위치를 파악한다. 위성이 얼마나 떨어져있는가를 계산함으로써 수신기는 스스로가 위성을 중심으로 하는 가상의 원공간 안에 있음을 알 수 있고 각 위성이 만들어내는 몇 개의 원공간 크기를 계산해내고 수신기는 이 원공간 들이 겹치는 부분에 있음을 알 수 있게 된다. 기본적으로 위치 측정은 GPS수신기가 3개 이상의 위성으로부터 정확한 시간과 거리를 측정하여 삼각 측량법에 의하여 현 위치를 정확히 계산 할 수 있으나 실제로는 위성의 위치, 시간오차, 대류층의 굴절이나 잡음신호 등이 정확한 위치 산출을 하는데 방해요인으로 작용하게 된다[1].

2.1.3 GPS의 정확도

GPS의 의한 정확한 위치측정은 수신기의 타입에 따라 달라진다. 대부분 휴대용 GPS 장비들은

10~20m 오차범위를 갖는다. DGPS(Differential GPS) 라고 하는 시스템은 훨씬 정교한 측정치를 보여주는데 DGPS는 인근의 확정된 지점에 고정된 추가 수신기를 필요로 한다. 고정 수신기에 의해 측정된 결과는 이동하는 수신 장치에 의해 기록된 위치를 교정하는데 사용되며 1미터 이내의 오차범위를 갖는 정확성을 나타낸다. 위성항법장치가 사용되기 시작했을 때에는 비군사용 사용을 제한하기 위해 100미터 오차의 시차오류를 삽입했었는데 2000년부터 제한이 풀려 일반인도 정확한 위치정보를 이용할 수 있게 되었다[1].

2.2 소켓 프로그래밍

2.2.1 소켓의 정의

소켓은 TCP/IP를 이용하는 API로써 1982년 BSD(Berkeley Software Distribution) 유닉스 4.1에서 소개된 이후 대부분의 유닉스 계열의 운영체제에서 기본적으로 제공하고 있다. 소켓은 유닉스의 보급과 함께 널리 이용되기 시작하였는데 BSD 유닉스는 기능이 계속 보완되어 1994년에 BSD 4.4 Lite가 발표되었고 이것이 FreeBSD, NetBSD, BSD/OS 등의 기초가 되었다.

소켓은 소프트웨어로 작성된 통신 접속점이라고 할 수 있는데 네트워크 응용 프로그램은 소켓을 통하여 통신망으로 IP패킷을 송수신하게 된다. 소켓은 응용 프로그램에서 TCP/IP 계층을 이용하는 창구 역할을 하고 있으며 응용 프로그램과 소켓 사이의 인터페이스를 소켓 인터페이스라고 한다. 한 컴퓨터 내에는 보통 한 세트의 TCP/IP 가 수행되고 있으며, 네트워크 드라이버는 LAN카드와 같은 네트워크 접속 장치를 구동하는 소프트웨어를 말한다[4].

2.2.2 소켓 프로그래밍 개요

클라이언트-서버 통신 모델에서는 항상 서버 프로그램이 먼저 수행되고 있어야 하는데, 서버는 socket()을 호출하여 통신에 사용할 소켓을 하나 개설하고 이때 리턴 된 소켓번호와 자신의 소켓주소를 bind()를 호출하여 서로 연결시켜 둔다. 서버에서 bind()가 필요한 이유는 소켓번호는 응용 프로그램이 알고 있는 통신 창구 번호이고, 소켓주소는 네트워크 시스템이 알고 있는 주소이므로 이들의 관계를 묶어 두어야 응용 프로세스와 네트워크 시스템간의 패킷 전달이 가능하기 때문이다.

다음에 서버는 listen()을 호출하여 클라이언트로부터의 연결요청을 기다리는 수동대기모드로 들어가며, 클라이언트로부터 연결요청이 왔을 때 이를 처리하기 위하여 accept()를 호출한다. 서버는 accept() 시스템 콜에서 기다리고 있다가 클라이언트가 connect()를 호출하여 연결요청을 해오면 이

를 처리한다. 이 때 연결이 성공적으로 이루어지면 서버와 클라이언트가 데이터를 송수신할 수 있게 된다.

한편 클라이언트는 socket()을 호출하여 소켓을 만든 후 bind()를 부를 필요 없이, 서버에게 연결 요청을 보내기 위하여 connect()를 호출한다. 이 때 클라이언트는 접속할 상대방 서버의 소켓주소 구조체를 만들어 connect()의 함수 인자로 주어야 한다[4].

3. 설계 및 구현

3.1 개발 환경

3.1.1 임베디드 보드

본 연구개발에서 사용한 임베디드 보드는 GPS 모듈이 장착되어 있는 HUINS사의 Acumen270K 보드를 사용하였다. 표 1은 사용된 임베디드 보드의 사양을 나타낸다.

표 1 임베디드 보드 사양

Item	Specifications
CPU	Intel XScale PXA27x: 520MHz
SDRAM	128Mbyte
Flash ROM	32Mbyte Flash
CF Card	50Pin Compact Flash 16bit CF Card
Ethernet	10/100Base T External Ethernet Controller (LAN91C111)
Serial	FFUART
USB 1.1	USB Host, USB Client
TFT LCD	7" (800×480) TFT LCD Touch Screen, Back Light 내장
Stereo I/O	Internal AC'97 Port, CS4202 Codec
User Define I/O	Switch Input 2EA, LED Output 2EA
Debugger	10Pin, 20Pin JTAG Connector

3.1.2 서버 PC

본 연구개발에서는 응용프로그램과 데이터베이스의 구현이 가능하도록 적당한 성능의 서버 PC를 선택하였다. 표 2는 사용된 서버 PC의 사양이다.

표 2 서버 PC 제품사양

Item	Specifications
CPU	Pentium 4 3.0GHz
MEMORY	1Gbyte
OS	Windows XP SP2
TOOL	Eclipse 3.3.2 한글판 JDK 6.0
Database	Oracle 10g

3.2 전체 구성

임베디드 보드인 클라이언트 개발과 서버의 데이터베이스 및 프로그램 개발로 파트를 나누어 구현을 하였다.

먼저 임베디드 보드의 기본 설정 및 HOST PC의 개발 환경을 구축하고, GPS모듈을 통해서 현재 위치를 전송 받는다. 전송받은 데이터는 소켓프로그래밍을 통해 보드에서 서버로 위치좌표를 전송하게 된다. 서버는 연결과정 프로그래밍 후 서버 프로그램의 개발 순으로 진행되었으며, 기본 OS는 Windows XP를 사용하고, 데이터베이스는 Oracle 10g를 사용하였다. 임베디드 보드로부터 전송받은 데이터는 데이터베이스에 저장되고, 저장된 위치 데이터와 지도와의 매핑은 Java언어를 통해서 구현하였다. 지도에서는 미아의 현재 위치를 볼 수 있으며, 데이터베이스에 저장된 데이터를 통해서 미아의 이전 궤적도 볼 수 있도록 구성하였다.

3.3 GPS 수신

임베디드 보드에 장착된 GPS모듈을 이용해 위성으로부터 현재 좌표를 수신 받을 수 있도록 코드를 작성하였으며, 아래 부분에 중요 코드 부분을 설명하였다.

3.3.1 GPS 수신 코드 분석 및 구현

GPS모듈은 PXA270의 STUART(GPIO<46,47>)에 연결하였다. 리눅스에서 STUART를 액세스하기 위해서는 /dev/ttySA2를 액세스 하면 된다.

우선 GPS포트를 OPEN해주고 그에 따른 에러를 처리하였다. 그리고 GPS포트를 통해 좌표를 수신받기 위한 기본 환경설정을 해주는 부분이 있으며, read()함수를 이용해 GPS좌표를 수신 받아서 receive_buff변수에 저장할 하게 된다[2].

3.3.2 GPS 수신 결과

그림 1처럼 수신된 좌표값들은 receive_buff 변수에 저장되고, 화면에 출력된다.

```

$GPRGA,023756.00,3751.9966,N,12744.3361,E,1.04,2.56,-00021,M,021,M, #42
$GPDZA,023757.44,15.08,2008. , #64
$GPRGA,023757.00,3751.9966,N,12744.3361,E,1.04,2.56,-00021,M,021,M, #43
$GPDZA,023758.43,15.08,2008. , #60
$GPRGA,023758.00,3751.9966,N,12744.3361,E,1.04,2.56,-00021,M,021,M, #4C
$GPDZA,023759.45,15.08,2008. , #6B
$GPRGA,023759.00,3751.9966,N,12744.3361,E,1.04,2.57,-00021,M,021,M, #4C
$GPDZA,023800.43,15.08,2008. , #6E
$GPRGA,023800.00,3751.9966,N,12744.3361,E,1.04,2.57,-00021,M,021,M, #4F
$GPDZA,023801.43,15.08,2008. , #6F
$GPRGA,023801.00,3751.9966,N,12744.3361,E,1.04,2.57,-00021,M,021,M, #4E
    
```

그림 1 GPS수신 결과화면

```

GPZDA,083022.43,27,10,2008. , #64
$GPRGA,083022.00,3752.0285,N,12744.2967,E,1.04,4.21,-00021,M,021,M, #4B
$GPDZA,083023.44,27,10,2008. , #62
$GPRGA,083023.00,3752.0285,N,12744.2967,E,1.04,4.21,-00021,M,021,M, #4A
$GPDZA,083024.27,10,2008. , #66
    
```

그림 2 클라이언트 전송 결과화면

그림 1의 값은 강원대학교 공대 6호관 연구실에서 좌표를 받은 수치로서 좌표의 값은 3751.9966,N,12744.3361,E에서 크게 벗어나지 않는다. 하지만 날짜와 시간은 UTC 시간으로서 한국 시간에 맞추려면 9시간을 더해주면 현재 한국 시간을 알 수 있다.

3.4 소켓을 이용한 좌표 전송

좌표를 전송하기 위해 서버와 클라이언트를 구축하였다. 서버는 응용프로그램이 개발된 컴퓨터로써 Windows XP 환경이며, 클라이언트는 임베디드 보드로써 Linux 환경이다.

3.4.1 리눅스 클라이언트

3.4.1.1 클라이언트 프로그램 구현

(1) 클라이언트 소켓 생성

서버의 소켓의 생성되어서 준비상태에 있다면 클라이언트도 소켓을 생성해서 서버에 접속할 준비를 해주어야 한다. 소켓이 생성되지 않는다면 error_handling()함수를 통해서 socket error가 발생한다.

(2) 서버와 연결

클라이언트를 서버에 연결하기 위하여 connect()함수를 사용하였으며 연결이 안된다면 error_handling()함수를 통해서 connect error가 발생한다.

3) 좌표 전송

GPS좌표를 저장하고 있는 receive_buff 변수를 소켓에 실어서 서버에 전송하는 코드이다. 우리가 필요로 하는 GGA좌표만 전송하기 위한 처리를 해주었으며 전송된 좌표는 화면에 출력 된다[3].

3.4.1.2 클라이언트 전송 결과

그림 2와 같이 서버 IP주소와 포트 번호를 입력하면 우리가 필요로 하는 GGA좌표만 서버에 전송됨을 확인할 수 있다.

3.4.2 윈도우 서버

3.4.2.1 서버 프로그램 구현

클라이언트와 마찬가지로 서버도 소켓을 생성해 주어야 한다. 또한 bind()를 통해서 자신의 소켓주소와 소켓번호를 연결시켜준다. 마찬가지로 bind가 안된다면 bind error가 발생한다. 그리고 listen()을 통하여 클라이언트로부터 연결요청을 기다리는 대기모드로 들어간다. accept()는 시스템 콜에서 기다리고 있다가 클라이언트가 connect()를 호출하여 연결요청을 하면 이를 처리해 준다. 그리고 recv()를 통해서 클라이언트로부터 데이터를 수신 받는다. 수신 받은 데이터는 message에 저장되게 된다. 이 데이터는 strtok을 통해서 gps[]변수에 “,”를 구분자로해서 토큰단위로 저장된다. message에 저장된 데이터는 시간과 좌표, 그밖에 값들이 모두 저장된 데이터이고, 이 중에서 우리가 필요로 하는 값들만 추출하기 위해서 토큰단위로 구분한다[3].

3.4.2.2 서버 수신 결과

클라이언트로부터 전송된 GGA좌표가 그림 3과 같이 화면에 출력된다. 이로써 보드에서 GPS모듈을 통해 좌표를 전송받고, 이를 소켓을 통해 클라이언트에서 서버로 전송을 완료하게 된다.

```

D:\source\server_win\Debug>echo server_win.exe 1004
$GPRGA 083022.00 3752.0285 N 12744.2967 E 1.04 4.21 -00021 M 021 M #4B
$GPDZA 083023.00 3752.0285 N 12744.2967 E 1.04 4.21 -00021 M 021 M #4A
$GPRGA 083024.00 3752.0285 N 12744.2967 E 1.04 4.21 -00021 M 021 M #4D
$GPDZA 083025.00 3752.0285 N 12744.2967 E 1.04 4.21 -00021 M 021 M #4C
$GPRGA 083026.00 3752.0285 N 12744.2967 E 1.04 4.21 -00021 M 021 M #4E
$GPDZA 083027.00 3752.0285 N 12744.2967 E 1.04 4.21 -00021 M 021 M #4F
$GPRGA 083028.00 3752.0285 N 12744.2967 E 1.04 4.22 -00021 M 021 M #4I
$GPDZA 083029.00 3752.0285 N 12744.2967 E 1.04 4.22 -00021 M 021 M #4D
$GPRGA 083030.00 3752.0285 N 12744.2966 E 1.04 4.21 -00021 M 021 M #49
$GPDZA 083031.00 3752.0285 N 12744.2966 E 1.04 4.21 -00021 M 021 M #48
$GPRGA 083032.00 3752.0285 N 12744.2966 E 1.04 4.21 -00021 M 021 M #4B
$GPDZA 083033.00 3752.0285 N 12744.2966 E 1.04 4.21 -00021 M 021 M #4A
$GPRGA 083034.00 3752.0285 N 12744.2966 E 1.04 4.22 -00021 M 021 M #4D
$GPDZA 083035.00 3752.0285 N 12744.2965 E 1.04 4.22 -00021 M 021 M #4F
$GPRGA 083036.00 3752.0285 N 12744.2965 E 1.04 4.22 -00021 M 021 M #4C
$GPDZA 083037.00 3752.0285 N 12744.2966 E 1.04 4.22 -00021 M 021 M #4D
$GPRGA 083038.00 3752.0285 N 12744.2966 E 1.04 4.22 -00021 M 021 M #42
$GPDZA 083039.00 3752.0284 N 12744.2966 E 1.04 4.22 -00021 M 021 M #42
$GPRGA 083040.00 3752.0284 N 12744.2965 E 1.04 4.22 -00021 M 021 M #4F
$GPDZA 083041.00 3752.0284 N 12744.2965 E 1.04 4.22 -00021 M 021 M #4E
$GPRGA 083042.00 3752.0284 N 12744.2965 E 1.04 4.22 -00021 M 021 M #4D
$GPDZA 083043.00 3752.0284 N 12744.2965 E 1.04 4.22 -00021 M 021 M #4C
$GPRGA 083044.00 3752.0284 N 12744.2965 E 1.04 4.22 -00021 M 021 M #4B
$GPDZA 083045.00 3752.0284 N 12744.2965 E 1.04 4.22 -00021 M 021 M #4A
    
```

그림 3 서버 수신 결과화면

3.5 데이터베이스 설계 및 구현

서버에 설치되어 있는 데이터베이스는 oracle 10g를 사용하며 클라이언트의 위치좌표들을 저장한다. 데이터베이스는 데이터 검색만 가능한 finder 계정과 데이터백업과 수정, 삭제가 가능한 scott 계정으로 분리하여 권한을 주도록 구현하였다.

서버가 동작을 시작하면 데이터베이스와 데이터를 받는 응용프로그램이 실행된다. 독립적으로 실행되는 서버 ODBC프로그램이 실시간으로 전송되어지는 좌표를 데이터베이스에 저장하고, Application은 데이터베이스에 접속하여 데이터검색이 가능하게 두 프로그램을 분리하였다.

임베디드 보드에서 받은 데이터는 콤마 단위로 구분하여 데이터를 그대로 데이터베이스의 테이블에 저장한 후에 응용 프로그램에서 이 데이터를 가공하여 사용한다[5]. 표 3에 데이터 베이스의 테이블 정보를 나타내었다.

표 3 테이블 정보

	자료형	설명
idx	NUMBER	인덱스
아이디	VARCHAR2(15)	클라이언트
시간	VARCHAR2(12)	시간자료
경도	VARCHAR2(10)	경도좌표
경도방향	VARCHAR2(2)	방향값
위도	VARCHAR2(10)	위도좌표
위도방향	VARCHAR2(2)	방향값

3.5.1 좌표를 데이터베이스에 저장

클라이언트로부터 수신된 데이터를 ID, 시간, 경도, 위도, 경도문자(N, S), 위도문자(E, W)로 필드를 나누어 데이터베이스에 저장한다[6]. 그림 4는 데이터베이스에 저장된 좌표의 예를 나타낸다.

아이디	시간	경도	위도	위
\$GPGGA 12744.6115	083006.998 E	3752.2235	12744.6115	N
\$GPGGA 12744.6115	083006.998 E	3752.2235	12744.6115	N
\$GPGGA 12744.6115	083006.998 E	3752.2235	12744.6115	N

그림 4 데이터베이스에 저장된 좌표

3.6 Application 프로그램의 동작

응용 프로그램의 동작은 크게 클라이언트에서

gps모듈을 이용하여 미아의 좌표를 전송받은 데이터베이스와의 연동을 통하여 미아의 현재위치를 확인하고, 해당 클라이언트의 이동경로를 파악하고, 마지막으로 클라이언트의 이동경로를 검색하는 기능을 가지고 있으며 모든 위치는 GPS좌표를 해당 지도 매핑 하여 나타낸다. 세부적인 기능으로는 지도 확대 및 축소, jxl을 이용한 엑셀파일 저장, 이미지 저장 등이 있다.

3.6.1 관리자모드 로그인

초기에 응용 프로그램을 구동 시키면 그림 5와 같은 초기 화면이 생기고 우측 하단에 로그인 창을 통하여 로그인이 가능하다.



그림 5 응용 프로그램 초기 구동 화면

3.6.2 로그인 후 데이터베이스와 연결

로그인 후에는 자동으로 데이터베이스에 연결되고 미아의 선택에 따라 데이터베이스에서 데이터를 가져오게 된다. 그림 6은 로그인 후 초기 화면을 나타내고, 그림 7은 미아 선택 메뉴를 나타낸다.



그림 6 관리자모드 로그인 후 화면(강원대학교 춘천캠퍼스 지역)

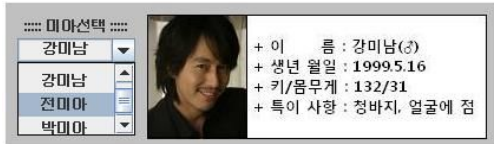


그림 7 미션선택 메뉴



그림 10 이동경로보기메뉴

3.6.3 현재위치 확인 및 이동경로 파악

그림 8과 그림 9의 현재위치보기와 이동경로보기 메뉴를 이용하여 맨 왼쪽부터 시작, 일시정지, 정지, 확대, 축소 등의 버튼을 이용하여 프로그램을 구동 시킨다. 먼저 재생버튼을 누르면 데이터베이스에 있는 해당 클라이언트의 GPS좌표를 가져와서 일련의 변환과정을 거친 후 클라이언트의 위치가 해당 지도에 표시되게 되고 일시정지와 정지버튼을 통해 위치 추적을 멈추거나 종료할 수 있다.



그림 8 현재위치보기 메뉴

현재위치는 데이터베이스에 실시간으로 데이터가 저장되기 때문에 현재시간과 비교해서 가장 가까운 데이터를 보여주게 된다.



그림 9 이동경로보기 메뉴

이동경로는 클라이언트의 궤적을 표시하는 기능으로 현재 날짜의 클라이언트의 이동경로를 모두 검색해서 표시한다.

3.6.4 이동 경로 검색

아래 그림 10의 이동경로보기메뉴에서 시작시간 및 종료시간을 조절하여 검색을 누르면 그림 11에서처럼 프로그래스바를 이용하여 패치정도를 알려주고 좌측에 있는 표에 시간과 각 클라이언트의 위치를 기록한다.



그림 11 이동경로 검색결과

3.6.5 지도내 좌표 매핑(Coordinates Mapping)

Coordinates Mapping은 지도 내 좌표설정으로 이번 시스템을 개발하면서 무선인터넷 환경 내에서 클라이언트에서 데이터베이스로 보내진 GPS좌표를 응용 프로그램에서 해당 지도에 표시하기 위해 사용되었다.

먼저 GPS에서 받아오는 여러 정보 중에 위도와 경도, 그리고 시간 이렇게 3가지 정보만을 사용하게 되는데 구글어스에서 제공하는 해당 지도의 위도와 경도의 최대, 최소값과 실제 지도의 가로, 세로의 최대, 최소 픽셀 값의 비를 구하여 실제 위도와 경도를 픽셀단위로 나타내었다.

각 클라이언트의 실제 좌표를 수신하여 이를 화면에 표시하기 위해서는 별도의 계산과정을 거쳐야 한다[10]. 계산과정에서 사용된 주요 변수 및 계산과정을 표 4에 나타내었다.

표 4 프로그램 내 주요 변수

m_start_x	화면지도의 시작 좌표	client_x	클라이언트의 위치좌표
m_start_y		client_y	
point_x	화면의 특정 위치	base_x	픽셀 당 좌표의 크기
point_y		base_y	
fct	확대/축소 기준		

공식 1 : GPS 좌표를 화면의 좌표로 변환

$$\begin{aligned} & (client_x - m_start_x)/(base_x/fct) & (1) \\ & (m_start_y - client_y)/(base_y/fct) \end{aligned}$$

공식 2 : 화면의 특정 위치를 실제 좌표로 변환

$$\begin{aligned} & m_start_x + (point_x * base_x / fct); & (2) \\ & m_start_y - (point_y * base_y / fct); \end{aligned}$$

3.6.6 세부 기능

응용 프로그램에서 추가적인 기능으로 클라이언트의 이동경로를 이미지파일과 엑셀파일로 기록을 남겨서 관리자가 이전 데이터를 찾기 용이하게 하였다. 케적이 남은 이미지 파일은 미아의 이동 모습을 한눈에 파악이 가능하다.

수집된 데이터는 일정기간이 지나면 테이블에 데이터를 XML파일로 백업 후에 테이블을 초기화시킨다. 데이터 백업단위는 1일을 기준으로 놀이동산의 개장과 폐장 시간사이의 위치데이터를 저장할 수 있도록 설계 하였다.

4. 결론

LBS 시스템은 위치 기반의 서비스로서 현재 많은 응용 서비스가 개발되고, 또한 제공 되어 지고 있다. 이러한 LBS 서비스 시스템의 중요한 응용으로서 본 논문에서는 임베디드 보드와 데이터베이스를 연동시켜서 미아 위치 정보 시스템을 개발 구현하였다.

본 논문에서의 구현은 GPS 신호 수신 및 위치 관련정보를 처리하는 임베디드 보드 부분, 여러 위치 정보를 저장하고 통합 관리하는 서버부분 및 이를 미아 위치 정보시스템의 기능을 제공하는 응용 어플리케이션 부분으로 나누어 진행되었다. 본 개발된 시스템은 Coordinates Mapping을 통해 GPS좌표를 해당지도에 나타낼 수 있으며, 세부적인 기능으로는 해당 좌표의 지도 매핑과 지도 확대 및 축소, jxl을 이용한 엑셀파일 저장, 이미지 저장 등이 있다.

앞으로 LBS 시스템은 더 많은 분야에서 사용될 것으로 예상되며, 좀 더 정확하고 섬세한 서비스가 제공 될 것이다. 이런 추세에 힘입어 이번에 개발한 미아 위치 정보시스템은 단순한 개인적인 문제에서 사회적인 문제로까지 대두되고 있는 미아발생에 대한 해결에 큰 도움이 될 것으로 사료된다.

참 고 문 헌

[1] 송대훈, 김성원, 최무영, *PXA270을 이용한 임베디드 시스템*, (주) 휴인스, 2006.
[2] 성호준, 김기도, *마이크로 소프트웨어*, 소프트뱅크미디어, 2005.

[3] 윤성우, *熱血講義 TCP/IP 소켓 프로그래밍*, 프리렉, 2007.
[4] 김화중, *컴퓨터 네트워크 프로그래밍*, 홍릉과학 출판사, 2001.
[5] 김태근, *10g로 시작하는 오라클*, 프리렉, 2007.
[6] 홍형경, *뇌를 자극하는 오라클 프로그래밍*, 한빛 미디어, 2008.
[7] 김승현, *열혈강의 자바 프로그래밍 (Java Programming)*, 프리렉, 2006.
[8] 케이시 시에라, 서환수, *Head First Java (뇌 회로를 자극하는 자바 학습법)*, 한빛미디어, 2005.
[9] 윤성준, *Java 세상을 덮친 Eclipse(개정판)*, 인사이트, 2005.
[10] 류대석, *Eclipse 를 이용한 Java 프로젝트 실무*, 제이에스 팩토리(주), 2008.