

# Windows Mobile 운영체제 기반 위치 트래킹 시스템의 구현 †

## Implementation of a location-tracking system based on Windows Mobile operating system

이정훈, 박경린  
 Junghoon Lee, Gyung-Leen Park  
 제주대학교 전산통계학과  
 {jhlee, glpark}@cheju.ac.kr

### 요약

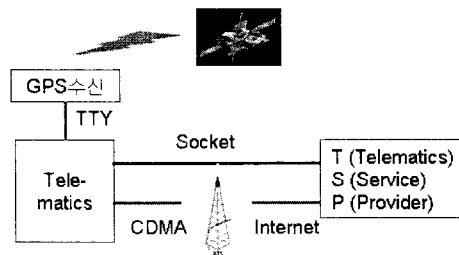
본 논문은 GPS 수신기와 CDMA 통신모듈을 장착한 이동단말기를 대상으로 자신의 위치를 Windows Mobile 운영체제의 API를 통해 중앙의 서버에게 전송하는 프레임워크를 설계하고 구현한다. 단말기 소프트웨어는 자원의 효율적인 사용과 확장성을 위하여 GPS 쓰레드, 통신 쓰레드 등으로 구성되어 비동기적으로 GPS 위치 정보를 공유 자료구조에 저장하고 응용에서 정의한 주기에 의해 중앙서버에게 보고한다. 서버는 이를 받아 위치 로그의 생성, 이동객체 위치 트래킹을 위한 가시화 등 다양한 응용을 작성할 수 있게 된다. Garmin 포맷의 분석, RAS에 기반한 CDMA 통신, 쓰레드 처리, 서버의 로그 생성 등 핵심적인 요소들의 구현에 의해 차량 텔레매틱스 등 다양한 위치기반 응용을 작성할 수 있도록 한다.

### 1. 서론

최근 GPS (Global Positioning System) 수신기를 장착한 휴대용 단말기가 널리 보급됨에 따라 위치에 기반한 서비스들의 개발에 대한 요구가 증가하고 있다. 예를 들어 건강 텔레매틱스 (health telematics), 차량 텔레매틱스 (vehicular telematics), 운송 등 다양한 서비스는 기본적으로 이동체의 위치를 추적하여 이에 대한 처리에 의해 서비스를 제공하는 프레임워크를 가진다[1]. 해당 단말기들은 최근 마이크로소프트의 Windows Mobile 운영체제를 탑재하고 있으며 이 운영체제는 단말기의 효율적인 자원관리와 아울러 전화망, 컴퓨터 네트워크 등에 대한 다양한 통신 프리미티브를 지원한다[2].

대부분의 차량 텔레매틱스 단말기들은 내비게이션 기능 지원을 목적으로 GPS 수

신기를 장착하고 있으며 WLAN 혹은 CDMA 모듈에 의해 텔레매틱스 장치간 혹은 텔레매틱스와 TSP (Telematics Service Provider)간 연결을 할 수 있다. 차량 텔레매틱스와 같은 이동성 단말기와 중앙 서버와의 연결은 (그림 1)에서 보는 바와 같이 GPS 수신 및 통신 모듈에 의해 구성되며 구성요소의 모듈별 구현과 개발은 전체 시스템의 효율화를 기할 수 있다[3].



(그림 1) 위치 트래킹 기본 프레임워크

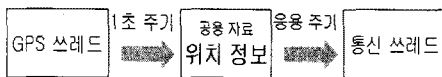
† 본 연구는 지식경제부 및 정보통신진흥연구원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음 (IITA-2008-C1090-0040)

더욱이 차량 텔레매틱스의 확산에 따라 Microsoft는 Windows Automotive를 출시하였는데 이 버전은 GPS 인터페이스에 관련된 API를 강화하였다[2]. 결국, GPS에 기반한 위치기반 응용들이 활성화될 것으로 예상되며 본 논문에서는 이러한 위치기반 응용들의 효율적인 구축과 수행을 위한 프레임워크를 설계하고 구현한다.

## 2. 프레임워크 구조

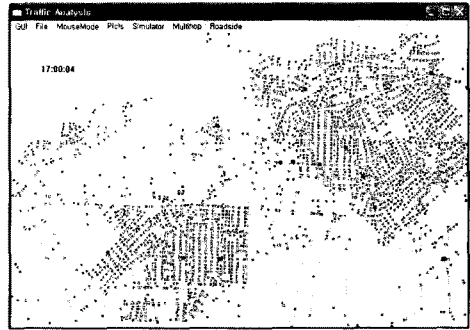
단말기의 소프트웨어는 GPS 인터페이스와 CDMA 인터페이스로 구성된다. GPS 인터페이스는 장치내의 직렬 포트를 통해 GPS 수신기와 연결되어 GPS 레코드를 수신하여 이를 분석하여 공용 자료구조에 저장한다. 통신 인터페이스는 시스템에서 정한 주기에 의해 이 자료구조를 읽어 중앙 서버에게 전달한다.

내장형 장치의 경우는 CPU나 메모리 자원의 성능이 일반 데스크탑에 비해 현저히 떨어지므로 자원의 효율적 사용을 위하여, 특히 입출력 장치에 관련된 기능을 수행하는 경우, 쓰레드 단위로 작업을 분할하는 것이 바람직하다. 작업분할은 (그림 2)에서 보는 바와 같이 GPS 쓰레드와 통신 쓰레드로 구성되었다. 물론 카메라 영상처리와 같은 새로운 쓰레드가 이 구조에 추가될 수도 있다.



(그림 2) 단말기 소프트웨어 구조

단말기가 음성 전달에 특화된 전화망으로 연결되는데 반해 서버는 데이터 처리에 중점을 두어 기본적으로 인터넷상에 위치한다. 서버의 입장에서는 단순히 Winsock 인터페이스를 통해 수신된 메시지를 처리하면 된다. 서버는 여러 이동체의 위치 정보를 수집하여 데이터베이스에 저장하거나 디지털 맵에 표시할 수도 있다[4]. (그림 3)은 수집된 위치 정보들을 디지털 맵에 표현한 예를 보이고 있다.



(그림 3) 수집 데이터의 디지털 맵 표현

## 3. 단말기 소프트웨어

텔레매틱스 단말기는 Windows Mobile 기반으로 개발이 되며 embedded Visual C나 Microsoft Visual Studio에 의해 각 응용이 작성된다.

### 3.1 GPS 쓰레드

GPS 수신기는 직렬 포트를 통해 단말기와 연결되어 있으며 프로그램에서 직렬포트로부터 데이터를 읽어야 한다. GPS 수신기로부터 읽혀진 정보는 (표 1)에서 보는 바와 같이 Garmin48 데이터 포맷을 따르는데 각 레코드는 \$로 시작되고 carriage return 문자로 레코드를 종료한다. RMC 레코드 이외에도 여러 가지의 레코드가 있지만 본 논문에서 대상으로 하고 있는 위치 결정에 대해서는 RMC 레코드만 처리하면 된다. 이 레코드는 시간, 타입, 경도(동경·서경 명시 포함), 위도(북위·남위 명시 포함), 노트로 표현된 속도, 진북을 기준으로 한 차량의 진행 방향 등을 포함하고 있다.

(표 1) Garmin48 레코드 구조의 예

\$RMC,070022,A,3327.197,N,12633.377,W,000.0,360.0,220800,007.2,W*78
\$RMC,070022,A,3327.233,N,12633.348,W,000.0,360.0,220800,007.2,W*78
\$RMC,070022,A,3327.197,N,12633.377,W,000.0,360.0,220800,007.2,W*78

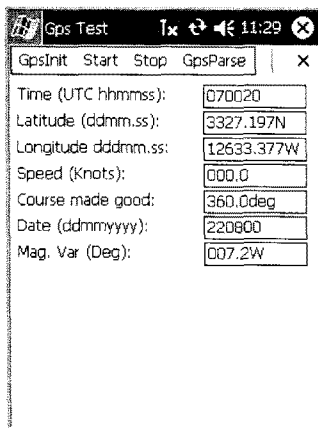
단말기로부터 GPS 수신기로 데이터를 보낼 수는 없으며 GPS 수신기는 1초마다

위치정보에 관련된 레코드를 사송으로 전송한다. 단말기 프로그램은 (표 2)에서 보는 것처럼 운영체제의 API 함수를 이용하여 직렬 포트를 열고 설치된 GPS 수신기에서 명세된 baud rate, 패리티, 정지 비트 수 등의 정보를 설정한다.

(표 2) 직렬 포트에서의 읽기

```
void FnGpsInit() {
    hGPSport = CreateFile (_T("COM5:"),
        GENERIC_READ | GENERIC_WRITE,
        0, NULL, OPEN_EXISTING, 0, NULL);
    // 직렬포트 설정
    dcb.BaudRate = CBR_4800;
    dcb.Parity = NOPARITY;
    dcb.StopBits = ONESTOPBIT;
    SetCommState(hGPSport, &dcb);
    // 한 바이트씩 읽어 문자열 구축
    while(hGPSport !=
        INVALID_HANDLE_VALUE) {
        ReadFile(hGPSport, &c, 1,
            &dwBytesRead, NULL);
    }
}
```

여기서 주의할 점은 GPS 수신기에서 PD A로 전송되는 레코드들은 문자당 1 바이트로 표현되는 ASCII 문자열의 형태인데 반해 PDA는 기본적으로 문자당 2 바이트로 표현되는 유니코드를 처리하므로 이에 대한 변환이 선행되어야 한다.



(그림 4) GPS 모듈 동작 예

Windows Mobile에서는 이에 대한 변환함수도 제공하고 있다. (그림 4)는 단말기에서 GPS 인터페이스를 통해 읽혀지고 분석된 위치 정보를 단말기 화면에 보인 예이다.

### 3.2 통신 쓰레드

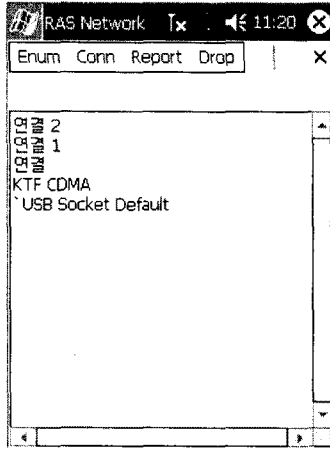
단말기에서 중앙의 위치 트래킹 서버와 통신을 하려면 연결된 통신 인터페이스에 해당하는 API를 이용하여야 한다. 단말기는 유선보다는 무선으로 연결되는 것이 일반적이며 각 응용의 목적에 따라 각각의 플랫폼마다 고유한 통신 인터페이스 구성을 갖는다. 물론 모든 통신 프리미티브에 있어서 Windows Socket이 가장 기본적인 기능을 제공하는데 이는 WLAN과 같은 컴퓨터 통신 네트워크 상에서만 가능하다.

(표 3) 통신회사 연결 정보의 조회

```
void FnRasEnum() {
    LPRASENTRYNAME lpRasEntry = NULL;
    DWORD dwRes, dwSize, dwEntries, dw
    lpRasEntry = new RASENTRYNAME[20];
    lpRasEntry[0].dwSize =
        sizeof(RASENTRYNAME);
    dwSize = sizeof(RASENTRYNAME) * 20;
    dwRes = RasEnumEntries (NULL, NULL,
        lpRasEntry, &dwSize, &dwEntries);
    for (dw=0; dw < dwEntries dw++) {
        PutEditWnd(lpRasEntry[dw].szEntryName);
    }
    delete[] lpRasEntry
}
```

CDMA (Code Division Multiple Access) 와 같이 전화망을 이용하는 경우는 RAS (Remote Access Service)에 의해 각 단말기가 중앙 서버에 연결할 수 있다. RAS 연결은 전화망 가입자가 사용하는 통신회사의 정보를 읽어서 연결을 요청하도록 되어 있으며 이러한 정보를 얻기 위해서는 (표 3)에서 보는 것처럼 Windows Mobile 운영체제의 RasEnumEntries API 함수를 호출한다. 이 함수를 호출한 결과

가 (그림 5)에 나타나 있으며 5개의 엔트리를 갖는다.



(그림 5) RAS Enumerator

이후에는 (표 4)에서처럼 실제로 RAS 연결을 통해 자신이 가입한 통신회사 게이트웨이로 전화 연결을 설정하게 된다. 이 코드에서는 KTF에 가입된 단말기 모듈임을 알 수 있으며 RasGetEntryParam 함수에 의해 해당 캐리어의 특성을 얻어온 후 RasDial 함수를 호출하여 통신회사와의 전화연결을 설정한다. 이 과정은 요금이 부과되며 교환되는 데이터의 양에 따라 요금이 결정된다. 데이터 통신 중에 전화 연결이 들어오면 자동으로 데이터 연결은 종료된다. 이는 하나의 단말기에 두 개의 통신 연결이 양립할 수 없기 때문이다.

RAS 연결이 설정이 되면 Winsock 함수를 통하여 일반적인 네트워크 응용처럼 서버와 통신할 수 있다. 즉, 클라이언트는 socket, connect, recv, send 등의 함수를 호출할 수 있다. 그러나 RAS 상에서의 socket 연결인 경우 단말기는 서버로는 동작을 할 수 없으며 오직 connect 함수에 의해 연결을 요청하는 클라이언트의 기능만을 담당할 수 있다. 따라서 서버가 먼저 데이터를 보내기 위해 연결을 설정하고자 하는 경우에도 서버는 클라이언트가 연결을 설정할 때까지는 데이터를 보낼 수가

없다. 결국 만약 보고 주기가 1분이라면 서버가 먼저 보내는 데이터의 경우 최대 1분까지의 지연시간을 갖는다.

(표 4) RAS 연결의 예

```
void FnRasConn(HWND hWnd) {
    RASDIALPARAMS rasDialParams;
    DWORD dwRes;
    BOOL bPassword;

    rasDialParams.dwSize =
        sizeof(RASDIALPARAMS);
    wcsncpy (rasDialParams.szEntryName,
        TEXT("KTF CDMA"));
    dwRes = RasGetEntryDialParams(NULL,
        &rasDialParams, &bPassword);
    dwRes = RasDial (NULL, NULL,
        &rasDialParams, 0xffffffff, hWnd,
        &g_hRasConn);
}
```

#### 4. 요약 및 결론

본 논문에서는 Windows Mobile 운영체제에 기반한 이동 단말기의 트래킹을 위하여 단말 및 서버의 소프트웨어 구조를 설계하고 그 요소들을 구현하였다. 본 논문에서 구현된 프레임워크와 요소기술들을 바탕으로 새로운 모듈들이 단말기에 추가될 수 있으며 이는 차량 텔레매틱스 등 다양한 서비스들이 효율적으로 제공될 수 있다.

#### 참고문헌

- [1] M. Imamura, K. Kobayashi, and K. Wat anabe, "Real-time positioning by fusing differential-GPS and local vehicle sensors," *SICE Annual Conference*, 2003.
- [2] <http://www.microsoft.com/windowsautomotive/default.aspx>
- [3] J. Lee, G. Park, H. Kim, Y. Yang, P. Kim, and S. Kim, "A telematics service system based on the Linux cluster," *LNC S*, Vol. 4490, pp. 660-667, 2007.
- [4] M. Grossniklaus, et al., "Putting location-based services on the map," *International Symposium on Web and Wireless Geographical Information System*, 2006.