

# S3C2410A와 Windows CE 5.0 기반의 임베디드시스템 개발에 관한 연구

김도규\*

## 목 차

- I. 서론
- II. RFID의 정의 및 응용사례
- III. SCM 글로벌 요인의 선행 연구
- IV. 통합된 SCM의 고객화
- V. RFID 한계점 과 유통산업의 향후방향
- VI. 실증연구
- VII. 결론
- 참고문헌
- Abstract

## I. 서론

현재 인터넷의 급속한 보급으로 일상생활에서 PC의 사용이 일반화 되어 있다. 최근에는 PDA, CNS, PMP, Handphone, 정보 가전기기 등 Post-PC 시대가 우리의 생활에 점차 가까워지고 있다. Post-PC 시대를 대표하는 핸드폰, PDA, PMP 등의 단말 장치는 크게 임베디드 시스템으로 분류할 수 있다. 다만 과거의 임베디드 시스템은 특정 목적을 위하여 다양한 기능이 불필요하였고 따라서 펌웨어 수준에서 모든 기능을 구현할 수 있었다. 그러나 최근에 임베디드 시스템은 고성능, 다기능의 SoC(System on Chip)를 기

반으로 사용자의 편의 제공 및 빠른 애플리케이션의 개발을 위하여 운영체제를 탑재하는 것이 자연스러워지고 있다.

최근에 임베디드 시스템을 위한 운영체제는 크게 3가지 계열로 구분되어 발전될 것으로 기대된다. ① 디스플레이 기능이 필수적으로 요구되는 계열로서 Windows CE 계열이 적합한 임베디드 시스템 ② 다양한 네트워킹 환경에서 정보통신 서비스를 제공하는 계열로서 모든 소스코드가 공개된 Linux 운영체제가 적합한 임베디드 시스템 ③ 엄격한 실시간 (hard realtime)을 요구하는 계열로서 상용의 RTOS가 필요한 임베디드 시스템.

본 논문에서는 PMP, CNS, PDA 등 디스플레이 기능이 필수적으로 요구되는 스마트 디바이스에 적합한 Windows CE 기반의 임베디드 시스

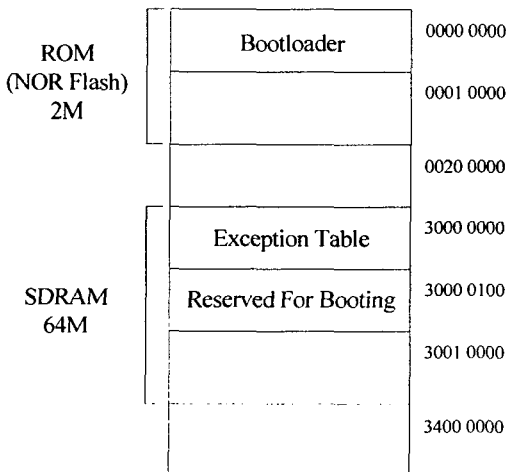
\* 성결대학교 정보통신공학부

템 개발에 관하여 연구하였다. 구체적으로는 ARM920T 코어 기반의 S3C2410A SoC를 사용한 개발 보드에 최근에 release된 Windows CE 5.0을 포팅하고 응용 프로그램으로 MP3 플레이어를 구현하였다. 2장에서는 Windows CE 기반의 임베디드 시스템 개발환경과 target 환경에 대하여 기술하고 3장에서는 Windows CE 기반 임베디드 시스템의 소프트웨어 구현 (BSP, CE 커널, 응용프로그램의 구현)에 대하여 기술하고, 결론은 4장에 기술하였다.

## II. Windows CE 5.0 기반의 임베디드 시스템 개발 환경

### 2.1. 시험용 보드의 하드웨어 구조

Windows CE 5.0 기반의 임베디드 소프트웨어가 탑재되는 시험용 보드는 ARM920T 코어 기반의 S3C2410A SoC 칩을 사용하여 개발된 보드이다. 시험용 보드의 메모리 맵은 <Fig. 1>과 같다.



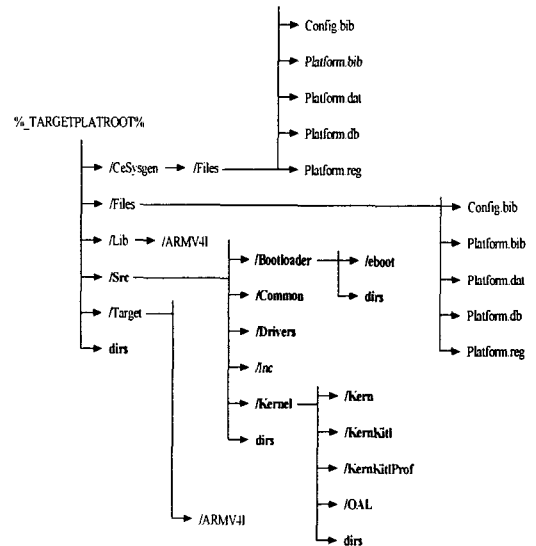
<Fig. 1> Target board의 memory map

### 2.2. Windows CE 5.0 개발 환경

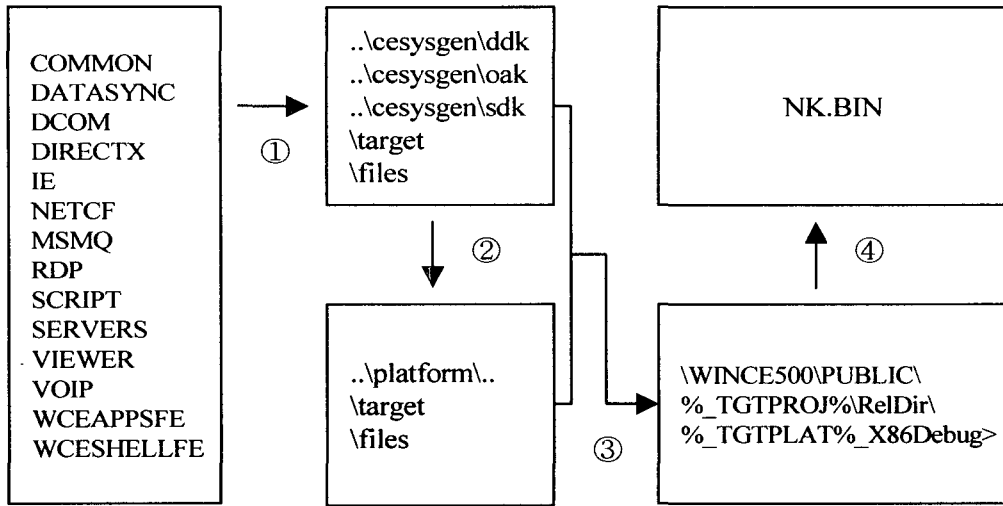
Windows CE에서 시험용 보드에 탑재되는 소프트웨어의 구성요소는 크게 BSP, Windows CE 커널, 응용프로그램이 포함되는 루트 파일시스템으로 구성되어 있고, 이들 소프트웨어의 개발을 위하여 PB (Platform Builder)와 eVC++ (eMbedded Visual C++) (or Visual Studio.NET)을 사용한다.

PB를 통하여 부트 로더와 CE커널의 개발, 빌드, 시험용 보드로의 다운로드 및 디버깅을 수행하고 또한 임베디드 응용 프로그램의 개발에 필요한 SDK (Software Development Kit)를 생성한다. PB에서 간단한 응용프로그램을 개발하는 것이 가능하지만 eVC++을 사용하여 PB가 생성한 SDK를 사용하여 응용프로그램을 개발하는 것이 더 효율적인 방법이다.

Windows CE 5.0 기반 임베디드 소프트웨어의 개발을 위한 플랫폼 구성은 <Fig. 2>와 같다.



<Fig. 2> Windows CE 5.0 기반 임베디드 소프트웨어의 개발을 위한 platform 구성



<Fig. 3 (a)> Build sequence

Build Procedure	수행하는 내용
① System generation	<p>* Sysgen.bat</p> <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; padding: 5px; width: 60%;"> <p>Header and source code files</p> <p>System libraries</p> <p>Module definitions (.def)</p> <p>Cesysgen.bat</p> </div> <div style="margin: 0 10px; text-align: center;"> <p>→</p> <p>sysgen.bat</p> </div> <div style="border: 1px solid black; padding: 5px; width: 35%;"> <p>System-specific header files</p> <p>Linked libraries</p> <p>Specific .def files</p> </div> </div>
② Feature Build	<ol style="list-style-type: none"> <li>1. Source code compile</li> <li>2. DIRS 파일 사용</li> <li>3. Sources 파일 사용</li> <li>4. nmake 호출</li> <li>5. 생성된 .exe와 .dll을 target 폴더로</li> <li>6. 생성된 .lib를 라이브러리 폴더로</li> </ol>
③ Building release directory	<p>* Buildrel.bat</p> <ol style="list-style-type: none"> <li>1. 모든 platform 파일들을 release directory에 복사</li> <li>2. 모든 binary 파일들을 release directory에 복사</li> <li>3. 사용자 설정에 의한 복사나 link 수행</li> </ol>
④ Making image	<ol style="list-style-type: none"> <li>1. release directory의 모든 파일들을 묶어서 nk.bin 이미지를 생성                         <ul style="list-style-type: none"> <li>- 모든 .bib 파일 =&gt; ce.bib</li> <li>- 모든 .reg 파일 =&gt; reginit.ini</li> <li>- 모든 .dat 파일 =&gt; initobj.dat</li> <li>- 모든 .db 파일 =&gt; initdb.ini</li> </ul> </li> <li>2. Default.fdf 생성</li> <li>3. nk.bin 이미지 생성</li> </ol>

<Fig. 3 (b)>

PB에서 시험용 보드를 위한 부트로더, 디바이스 드라이버, OAL, 커널, 응용프로그램을 포함하는 소프트웨어 이미지를 빌드하는 순서는 다음과 같다(<Fig. 3> 참조).

### III. Windows CE 5.0 기반 임베디드 시스템의 소프트웨어 구현

Target 보드를 위한 CE 기반의 임베디드 소프트웨어의 개발은 PB가 제공하는 서비스를 최대한 이용하여 개발 하였다. PB는 Catalog를 통하여 개발 보드에서 필요한 기능을 제공하는데 그 기능은 다음과 같다.

- ① BSPs : 다양한 prototype 보드에 대한 BSP 제공
- ② Core OS : target 보드에 탑재되는 커널의 다양한 feature를 제공
- ③ Device Drivers : target 보드의 디바이스에 대한 장치관리자 제공
- ④ Platform Manager : target 보드에 탑재되는 부분과 개발 호스트에 탑재되는 부분으로 구성되고 kernel connectivity와 application connectivity를 제공한다.
- ⑤ Third Party BSPs : target 보드의 reference 보드에 대한 BSP

Windows CE 기반으로 임베디드 소프트웨어를 개발하는 과정은 다음과 같다.

- ① Create BSP
- ② Create Platform
- ③ Compile Platform

- ④ Eboot Porting
- ⑤ Kernel OS Porting
- ⑥ Building SDK
- ⑦ Develop Application

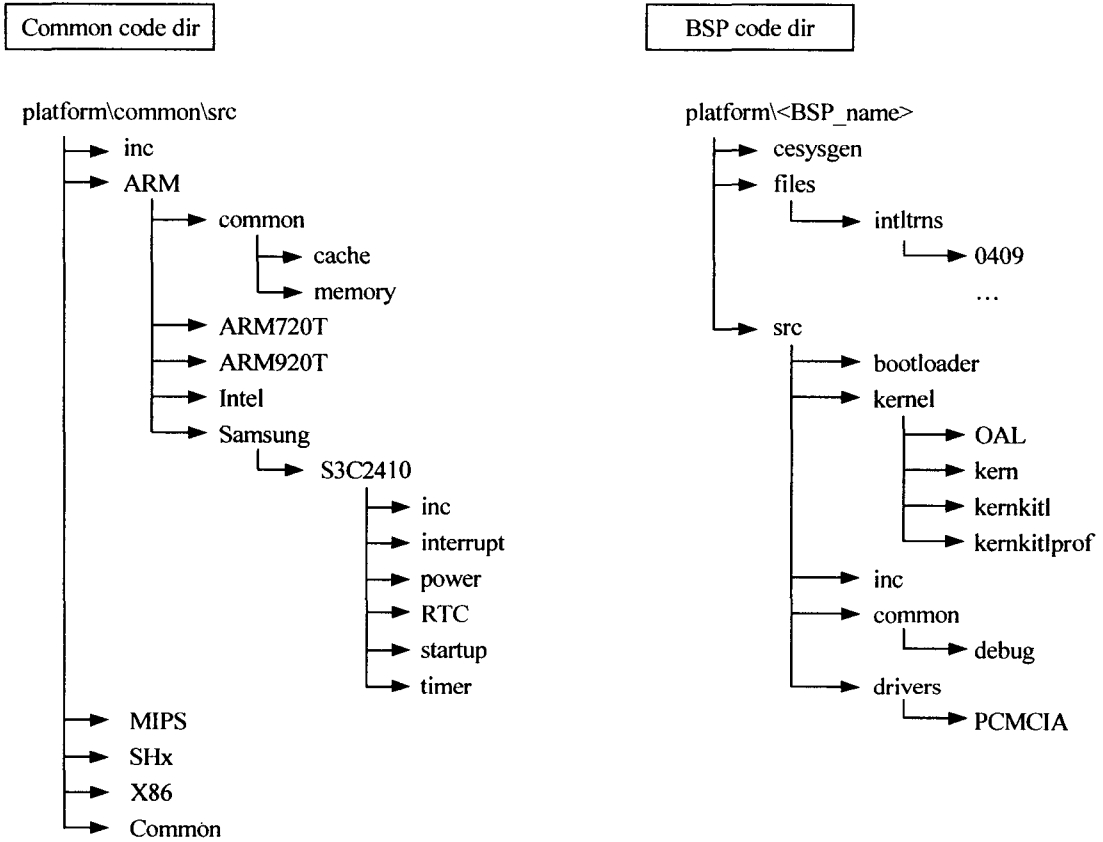
CE 기반 임베디드 소프트웨어의 핵심 요소는 3가지로 ① BSP (Board Support Package) ② 커널 ③ SDK와 Application로 구성되어 있다. 이 3가지 구성요소 가운데 BSP는 임베디드 시스템의 소프트웨어 개발에서 제일 중요한 부분으로 개발하려고 하는 보드에 적합하도록 개발되어야 하는 부분으로 ① 부트 로더 ② 디바이스 드라이버 ③ OAL(OEM Adaptation Layer, H/W에 의존하는 OS 코드)로 구성되어 있다. PB가 지원하는 CPU 환경과 BSP 개발을 위한 디렉토리 구조는 <Fig. 4>와 같다.

Platform 디렉토리에서 해당되는 파일을 빌드하기 위해 필요한 파일은 ① DIRS ② SOURCES로 구성되어 있다. 또한 개발 보드용 image 파일을 생성하기 위한 설정 파일은 ① .bib ② .reg ③ .dat ④ .db 로 구성되어 있다.

PB의 New Platform Wizard를 이용하여 개발 보드를 위한 platform을 생성하는 절차는 다음과 같다 (여기에서 platform이란 특정 개발 보드를 위한 Windows CE OS를 구체적으로 구현하는 것을 의미하고, 이것은 OAL과 디바이스 드라이버를 포함하는 BSP와 OS image로 구성된다).

- ① Chooset a BSP
- ② Choose a design template
- ③ Add items to OS design of developing board or remove items from it

개발하려는 임베디드 시스템의 커널은 원하는 application을 용이하게 개발할 수 있도록 PB의 카탈로그 목록에서 제공하는 다양한 컴포넌트를



(Fig. 4) BSP 구현을 위한 platform 구성

선별적으로 추가하여 구성한다. 특별히 Windows CE 5.0 기반의 임베디드 시스템의 어플리케이션 개발은 커널에서 제공하는 기능만을 사용하여 개발되기 때문에 커널의 최적화와 밀접한 관련이 있다.

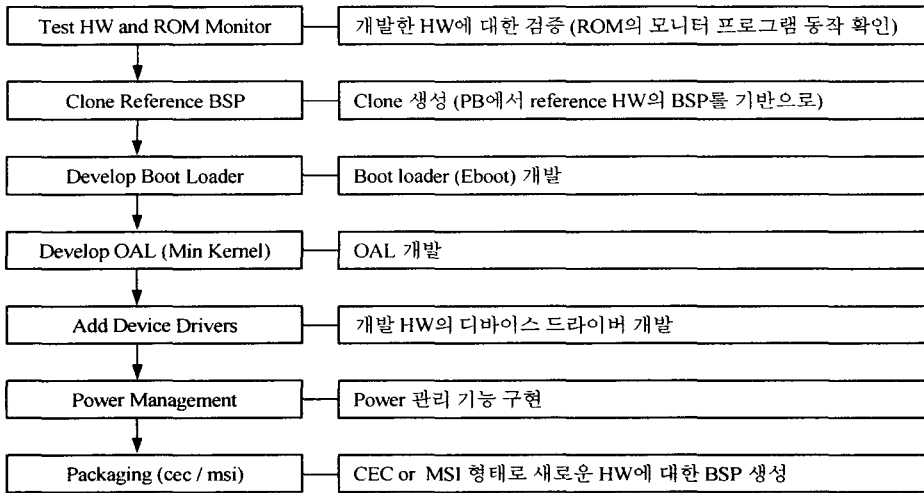
### 3.1. BSP 구현

개발 보드를 위한 BSP의 구현은 PB에서 제공하는 BSP Wizard 기능을 이용하여 구현하였다. PB에서는 ARM920T 기반의 S3C2410A를 사용하여 구현한 SMDK2410 보드에 대한 BSP를 제공하고 있다. 개발 보드 역시 S3C2410A를 사용

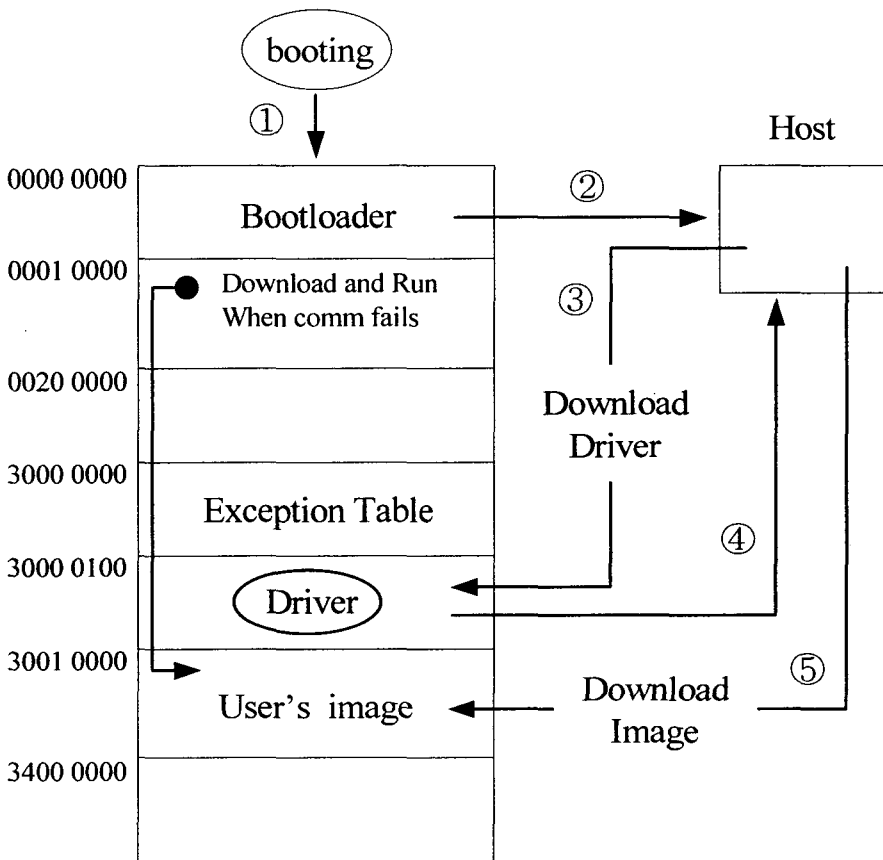
하고 있으므로 SMDK2410의 BSP를 기반으로 하여 개발 보드의 BSP를 개발하였다 (<Fig. 5> 참조).

<Fig. 6>은 BSP의 핵심 부분 가운데 하나인 부트로더가 초기 부팅 호스트의 다운로드 프로그램(ARMDown)과 통신을 시도하고, 통신이 되면 ARMDown을 통해 driver와 실행 이미지를 다운로드하는 과정을 보여주고 있다.

OAL을 구성하는 핵심요소의 하나인 부트로더는 Eboot이고 구현된 메뉴 내용은 다음과 같다.



<Fig. 5> BSP 개발과정



<Fig. 6>

// -----

Ethernet Boot Loader Configuration :

### Ethernet Setting ###

- 0) IP address : 211.221.245.140
- 1) Subnet mask : 255.255.255.0
- 2) DHCP : Enabled
- 3) Program CS8900 MAC address

### System Setting ###

- 4) Boot delay : 1 seconds
- 5) Startup: Download new image
- 6) NAND Write Protect : Disabled
- 7) Program RAM image into NAND Flash : Disabled
- 8) Low-level format the NAND Flash

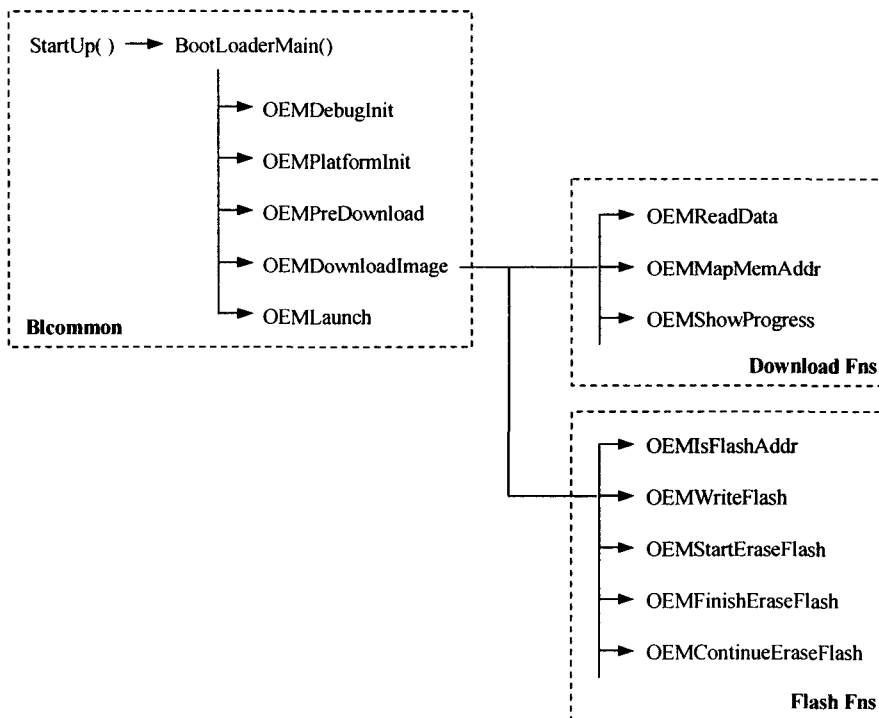
9) Reset to factory default configuration

- U) download image via USB
- E) download image via Ethernet
- L) Launch existing flash resident image now
- S) Save configuration

Enter your selection :

// -----

Eboot 코드의 동작 흐름은 <Fig. 7>과 같다. 부트 로더의 시작은 Blcommon의 StartUp()에서 시작하고 주 기능은 BootLoaderMain()에서 이루어지게 된다.



<Fig. 7> Boot loader의 제어 흐름

### 3.2. Kernel 구현

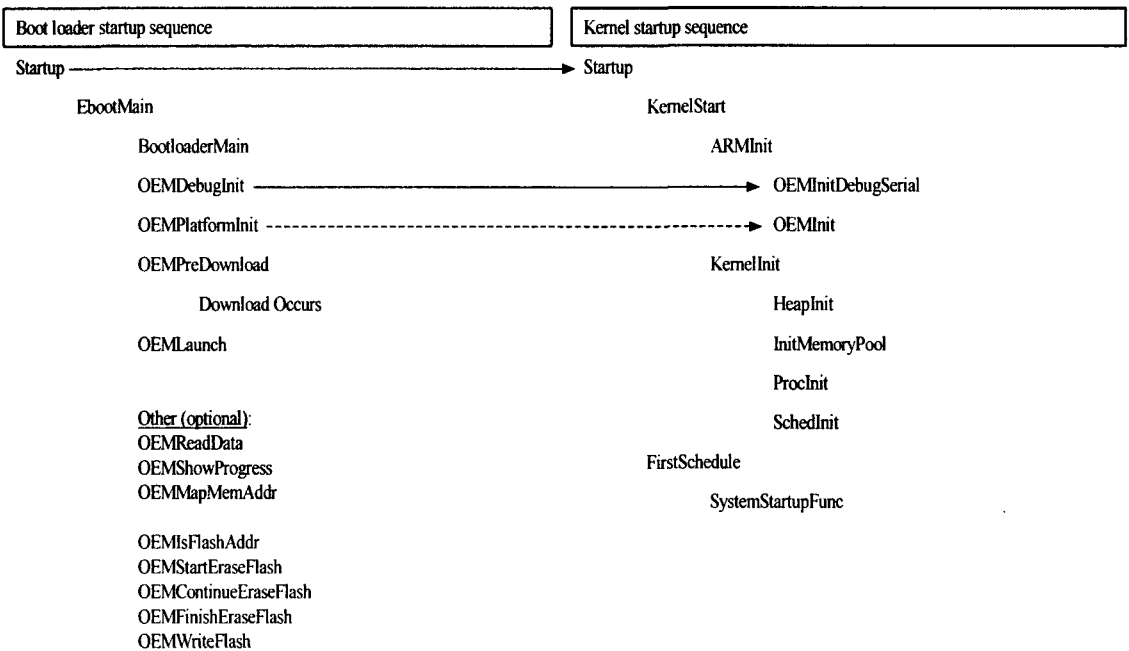
Target 보드를 위한 CE 커널은 PB의 카탈로그(catalog) 항목 중 Core OS에서 필요한 feature를 선택하여 프로젝트에 추가함으로써 용이하게 구성할 수 있다. PB의 Catalog | Core OS의 중요 feature는 다음과 같다.

- ① Applications and Services Development
- ② Applications - End User
- ③ Communication Services and Networking
- ④ Core OS Services
- ⑤ Device Management
- ⑥ File Systems and Data Store
- ⑦ Fonts
- ⑧ Graphics and Multimedia Technologies
- ⑨ International
- ⑩ Internet Client Services

- ⑪ Security
- ⑫ Shell and User Interface
- ⑬ Voice over IP Phone Services
- ⑭ Windows CE Error Reporting

개발 보드를 위한 커널의 구성요소는 platform wizard를 이용하여 프로젝트를 생성하면서 기본적인 feature가 추가되고 나중에 필요한 feature를 부가적으로 추가 시킬 수 있다. Mobile handheld를 위한 기본적인 요소는 다음과 같다.

- ① Applications - End User
- ② Applications and Services Development
- ③ Communication Services and Networking
- ④ Core OS Services
- ⑤ File Systems and Data Store
- ⑥ Fonts
- ⑦ Graphics and Multimedia Technologies



〈Fig. 8〉 Boot loader에서 커널로의 제어 흐름



- ⑧ International
- ⑨ Internet Client Services
- ⑩ Security
- ⑪ Shell and User Interface

<Fig. 8>은 BSP의 부트 로더에서 CE 커널로 제어가 넘겨지는 과정을 각 부분의 코드에서 진행되는 과정을 보여주고 있다.

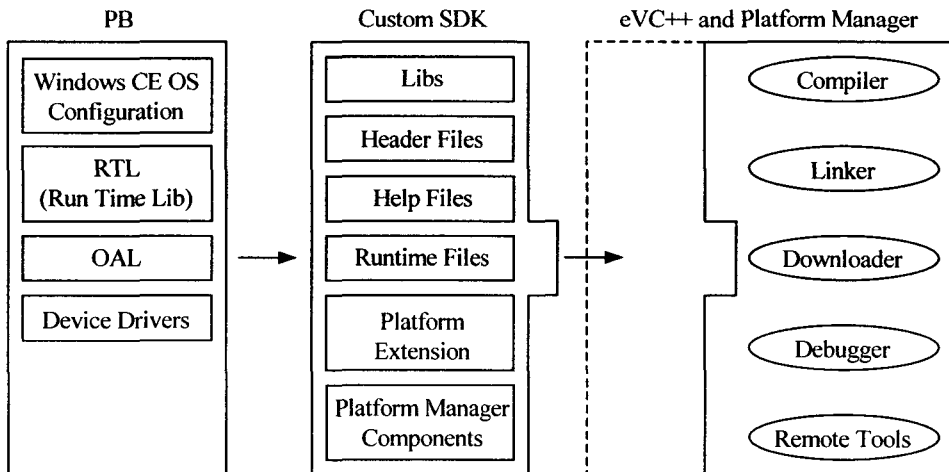
### 3.3. 응용프로그램 구현

Windows CE에서는 커널과 응용프로그램이 하나의 이미지 nk.bin으로 구현된다. 현재 CE 기반의 응용 프로그램 개발은 PB나 eVC++를 통하여 이루어진다. CE 기반 응용프로그램 개발이 일반 PC에서의 개발과 다른 점은 target 보드를 위한 SDK (Software Development Kit)의 구현이 선행되는 것이다. 이것은 각 target 보드에서 동작하는 커널의 기능이 서로 다를 수 있기 때문이다. 따라서 PB를 이용하여 커널이 제공하는 기능을 SDK로 구현하여 외부에 제공하고, 이것을 이용하여 응용 프로그램을 개발할 수 있다

(<Fig. 9>참조).

Sample application으로 MP3 player를 구현하였다. 그 과정은 다음과 같다.

- (i) PB에서 SDK를 구현하는 과정
  - ① New SDK in PB | SDK : 응용 프로그램을 위한 SDK의 생성을 진행한다.
  - ② Configure SDK in PB | SDK : SDK의 설정 작업을 수행한다.
  - ③ Build SDK in PB | SDK : 아래의 위치에 필요한 SDK가 저장된다.(%\_WINCEROOT%\PBWorkspaces%\\_PRJNAME%\SDK%\\_PRJNAME%\SDK.msi)
- (ii) eVC++에서 SDK를 사용하여 응용프로그램을 개발하는 과정
  - ① PB에서 빌드한 %\_PRJNAME%\SDK.msi를 설치한다. Default 설치 위치는 C:\Program Files\Windows CE Tools\wce500이다.
  - ② Run eVC++



<Fig. 9> Custom SDK 기능

③ Choose configuration in WCE Configuration Toolbar

④ Develop WCE application: 일반적으로 WCE 응용프로그램의 개발 시 주의 할 사항은 다음과 같다.

- Folder만 존재하고 상대 폴더는 존재하지 않는다.
- UNICODE를 기본으로 처리한다.
- 다양한 platform이 존재하고 각 platform에 다양한 CPU가 존재한다.

여기에서는 DirectShow를 사용하여 MP3 플레이어를 구현하였다. 그 기본 골격은 다음과 같다.

```
// -----
#include <streams.h>
#pragma comment(lib, "ole32.lib")
#pragma comment(lib, "strmiids.lib")

IGraphBuilder *pGB = NULL;
IMediaControl *pMC = NULL;

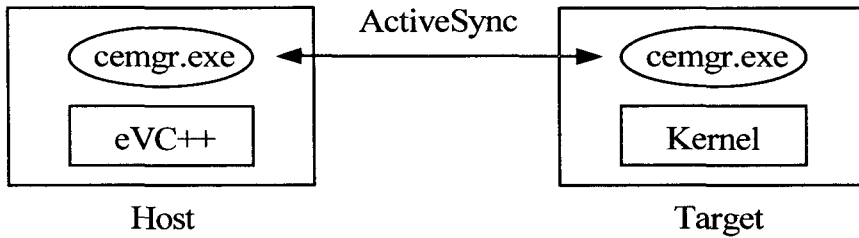
HRESULT PlayMP3(LPWSTR szFile)
{
    HRESULT hr;
    CoInitialize(NULL);
    hr=CoCreateInstance(CLSID_FilterGraph,
        NULL, CLSCTX_INPROC_SERVER, IID_
        IGraphBuilder, (void**) &pGB );
    if(SUCCEEDED(hr)) {
        pGB->RenderFile(szFile, NULL);
        pGB->QueryInterface(IID_IMedia
        Control, (void**) &pMC);
        pMC->Run();
    }
    return hr;
}
```

```
LRESULT CALLBACK MP3DlgProc( ... )
{
    switch (msg) {
        case WM_CHAR:
            {
                TCHAR chCharCode =
                (TCHAR) wParam;
                switch
                (chCharCode) {
                    case L'0' :
                        PlayMP3(L"\\test.mp3");
                        break;
                    case L'1' :
                        pMC->Pause();
                        break;
                    case L'2' :
                        pMC->Stop();
                        break;
                }
            }
    }
    return FALSE;
}
// -----
```

⑤ Build application

⑥ Run and debugging: 디버깅을 위해서 host-target의 연결을 관리해야 한다. 이때 target의 kernel에 platform manager가 포함되어 있어야 한다 (<Fig. 10> 참조).

<Fig. 10> Platform manager를 통한 host-target의 connectivity 관리



## V. 결론

본 논문에서는 ARM920T 코어 기반의 S3C-2410A SoC를 사용한 개발 보드에 최근에 release된 Windows CE 5.0 기반의 임베디드 시스템 개발에 대하여 연구하였다. 시험용 보드에 탑재되는 임베디드 소프트웨어는 ① BSP (Board Support Package) ② Windows CE 5.0 커널 ③ 응용프로그램으로 구성되어 있다. BSP의 핵심요소는 부트로더와 디바이스 드라이버 등으로서 특히 통합개발환경인 PB를 통하여 부트로더 및 OAL을 시험용 보드에 다운로드하고 디버깅하는 등의 기능을 효율적으로 수행하였다. Windows CE 5.0 커널은 카탈로그에서 제공하는 feature들을 적절히 선택하여 용이하게 구성할 수 있었다. 응용프로그램의 개발을 위해서 먼저 커널에서 제공하는 기능을 SDK로 구현하였고 이를 이용하여 간단히 MP3 응용프로그램을 개발하여 간단히 테스트하였다. 향후 PMP, CNS, PDA 등 디스플레이 기능이 필수적으로 요구되는 스마트 디바이스 등에는 Windows CE 기반으로 다수의 임베디드 시스템 개발이 더욱 활발해 질 것으로 기대된다.

## 참고문헌

- [1] <http://msdn.microsoft.com/embedded/windowsce/>
- [2] Mobile and Embedded Development at <http://msdn.microsoft.com/library/>
- [3] "Windows CE5.0 Review", Embedded World, 2004, No.9.
- [4] "OAL 개발방법에 관한 연구", 김도규, 성결대학교 정보기술연구소 논문집, 2005.
- [5] "Building of a Windows CE image", Dokyu Kim, Technical Report, 2005.
- [6] "Mobile PDA Programming", 고재관, 삼각형프레스, 2001.
- [7] "Inside ATL/COM", 전병선, 삼양출판사, 1998.

## Development of Embedded System Based on Windows CE 5.0

Do-Kyu Kim\*

### Abstract

In this paper, development of embedded system based on Windows CE 5.0 which released recently is studied. Embedded softwares for the target board using S3C2410A SOC based on ARM920T core are composed of ① BSP(Board Support Package) contains an OAL(OEM Adaptation Layer) which includes a boot loader for initializing and customizing target hardware, device drivers, and a corresponding set of configuration files ② Windows CE 5.0 kernel ③ SDK and MP3 test application. Particularly, PB(Platform Builder) provides the efficient functions to build, test and debug the BSP and CE kernel. It is looked forward to being widely spread that Windows CE 5.0 will be utilized at smart devices such as PMP, CNS and DMB phone which inevitably require a display device.

Key Words: Embedded System, Windows, PMP, CNS, DMB

---

\* Professor, Sungkyul University