

嵌入式系統的 BSP 與 Turn-Key 簡介

廖鵬宇 劉岱穎 林盈達

國立交通大學資訊工程系

E-MAIL : {pyliao, liuty, ydlin}@cs.nctu.edu.tw

October 20, 2008

摘要

開發板支援套件(Board Support Package, BSP)與統包(Turn-Key)在嵌入式系統中扮演無可取代的角色,透過 BSP 可讓作業系統順利載入並運行於硬體平台, BSP 也提供驅動程式讓硬體容易控制; Turn-Key 為事先編譯好的套件,套件中包含作業系統、工具鏈(Tool-Chain)與相關應用程式,將 Turn-Key 燒入系統,便可讓應用程式運行。嵌入式系統中 BSP 與 Turn-Key 為整個系統的軟體部分,包含核心、驅動程式與上層應用程式,使用 Linux 核心的 Turn-Key 可直接運行於 ARM 架構的開發板,加入 BSP 到核心更可讓開發板的硬體得到完整支援與控制。

關鍵字: Embedded Linux、WinCE、Board Support Package、Turn-Key

1. 簡介[1]

嵌入式系統(Embedded System)是一種「完全嵌入硬體設備,為特定應用設計的計算機系統」,與一般電腦不同,嵌入式系統為針對特定用途而設計,因此強調核心與軟體達到最佳化,其應用範圍包含 PDA 等手持式產品,電玩與 3C 產品也有嵌入式系統的應用,我們先將作業系統核心與需要的應用軟體編譯成映像檔,編譯完成後再燒入儲存體當中,燒入之後對於軟體不做太多更改,因此也將燒入的映像檔稱為「韌體」,其中的映像檔包含開機管理程式(Boot Loader)、系統核心(Kernel)、BSP、Turn-Key 與上層應用程式。

BSP 介紹[1][2]

BSP 的全名為 Board Support Package,它是一系列程式碼的集合,扮演硬體與作業系統溝通媒介的角色,能讓作業系統在開發板上運行,作業系統可藉由 BSP 所提供的界面,進而對硬體做進一步的控制。

以下為 BSP 的主要功能:

1. 硬體的初始化,主要是針對 CPU 做初始化,為上層軟體提供硬體支援。
2. 為作業系統提供設備驅動程式與系統中斷服務程式。
3. 定製作業系統功能,提供軟體一個多工的操作環境。

4. 初始化作業系統，讓作業系統能夠正常運行。

嵌入式系統最基本的硬體環境包含中央處理器(CPU)，隨機存取記憶體(RAM)和主要儲存體(通常為Flash Read-Only Memory)，BSP 至少要提供這三項元件的控制才能讓作業系統運作。

Turn-Key 介紹

Turn-Key 指的是一項產品的完整解決方案，例如安裝一個軟體，只要按照安裝步驟，便可以完成軟體安裝，而這指示步驟便是一種完整解決方案(Turn-Key Solution)。

在嵌入式系統當中，我們也可以將 Turn-Key 定義為事先編譯好的套件，包含 Kernel 的映像檔與根檔案系統(Root File System)的映像檔，使我們不用再建立開發環境另外進行測試、編譯等工作，便可直接將準備好的映像檔燒錄到嵌入式系統的儲存體當中。

嵌入式系統整體系統架構[1]

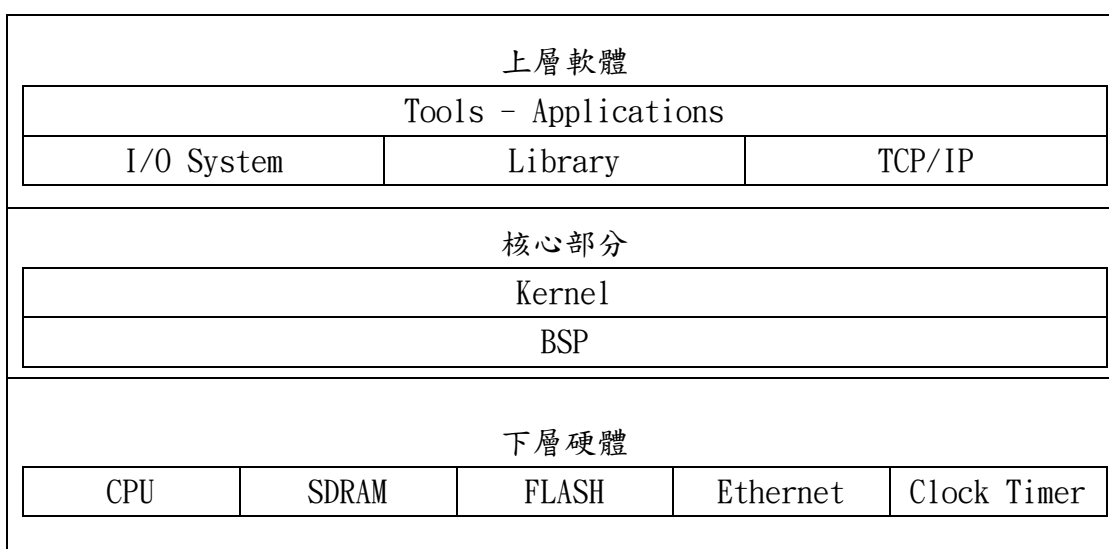


圖 1. 系統架構圖

由圖 1 可以了解 BSP 位於核心部分底層，介於 Kernel 與下層硬體之間，kernel 可透過BSP來控制下層硬體，而先前提到Turn-Key為事先編譯好的套件，在嵌入式系統中，這些事先編譯好的套件會以映像檔的方式呈現，主要的映像檔分別為 Kernel Image 與 Root File System Image。

Kernel Image 為系統架構圖的核心部分，裡面包含 kernel 的 source code 與硬體廠商提供的 BSP；而 Root File System Image 即圖 1 上層軟體的部分，裡面包含 Library、GCC Compiler 與一些應用程式等，整體來看，除了下層硬體

部分之外，其他都是 Turn-Key 所包含的範圍。

2. BSP 的開發過程[3][4]

和一般開發程式一樣，BSP 的開發也分成 High-Level 與 Low-Level 等層級的工作，Highest-Level 的部分包含如何去選擇硬體，Lowest-Level 的部分包含如何設計 Boot Loader 與 OEM 轉換層(OEM Adapter Layer, OAL)，圖 2 為 BSP 的開發流程。

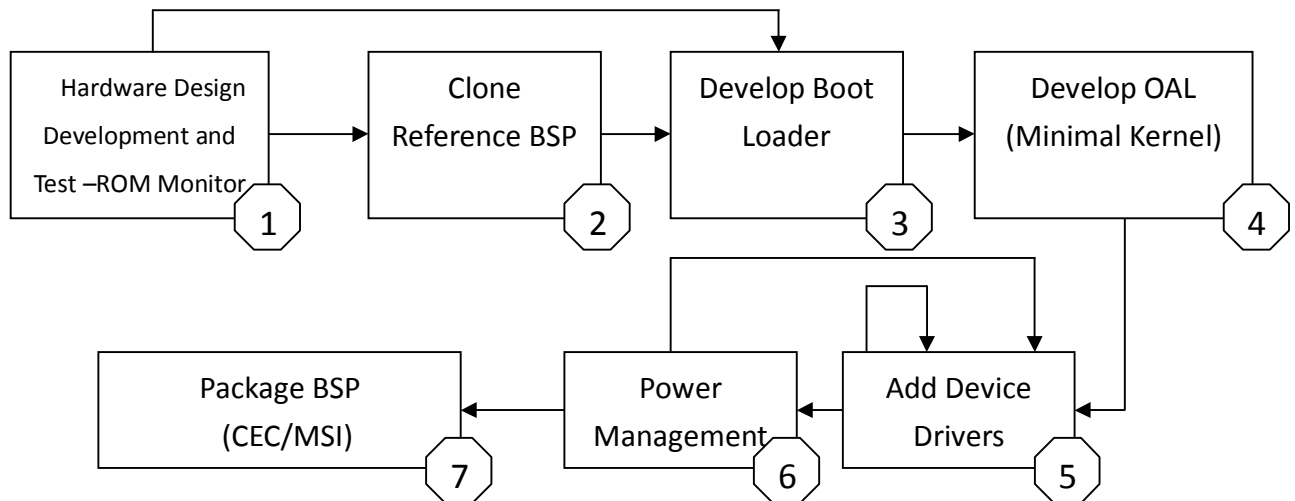


圖 2. BSP 開發流程圖[4]

設計 BSP 如圖 2(1)我們要選擇嵌入式系統所要執行的硬體平台，或是選擇自行設計，再針對選擇的平台建立 BSP，可以從圖 2(2)同一硬體架構下的 BSP 作修改，也可以自行設計 BSP，主要設計重點除了針對硬體平台而設計之外，也要考慮選用的作業系統，每種作業系統對硬體定義的函式皆不同，所謂的函式可視為一種函式呼叫，函式內容包含呼叫硬體的程式碼，作業系統要控制硬體只需透過這些函式即可控制硬體，找出函式名稱的定義，依據每個函式名稱來修改函式內容。

設計 BSP 一開始會先從圖 2(3)Boot Loader 做起，由初始化硬體並將核心載入記憶體執行，接下來圖 2(4)開發 OAL，OAL 具備啟動核心過程所需的功能，讓作業系統順利在硬體上運行，圖 2(5)在 BSP 中加入裝置驅動程式，使作業系統可以控制周邊的硬體設備，圖 2(6)加入電源管理的功能，我們會考慮在不同情況下使用不同電源模式，探討在不同模式中各個硬體元件運作效率，達到省電的功效。

圖 2(7)完成 BSP 後，依照作業系統不同，使用不同呈現方式，WinCE 會將 BSP 整理成資料夾的形式，而 Linux 會將 BSP 與 Kernel 原始碼放在一起，之後段落會針對 Linux 與 WinCE 的 BSP 作比較。

3. BSP 相關比較

BSP 在嵌入式系統與 X86 系統的不同

在 PC 當中，運行 Windows 或 Linux 也需要 BSP，不過這些作業系統都是運行在 x86 系統當中，因此不需特別修改 BSP 便能讓作業系統運行並控制硬體，而在嵌入式系統當中，由於使用的硬體平台有不同規格(ie. Power PC, ARM, MIPS 等)，指令集也有 RISC 與 CISC 的差別，使得每個嵌入式系統都有不同硬體架構，就算使用相同的硬體平台，每一個邏輯板也因週邊設備修改而不同(ie. RAM 的大小)，所以廠商除了嵌入式開發板外，也會針對開發板的硬體規格設計專屬的 BSP，讓開發者透過 BSP 控制整塊開發板的硬體，表 1 為嵌入式與 x86 平台的 BSP 比較整理。

比較物 項目	嵌入式的 BSP	X86 的 BSP
硬體相容性	不同硬體不可共用	可共用 BSP
作業系統相容性	不同作業系統不可共用	可共用 BSP
需修改 BSP	多	少

表 1. 嵌入式 BSP 與 x86 的 BSP 比較[1]

BSP 與 PC 上 BIOS 的區別

BSP 與 PC 上 BIOS 有很大的區別，BIOS 的主要任務為電腦開啟時的檢測、系統硬體的初始化(中斷向量表、記憶體等)、載入作業系統、讓作業系統對硬體發出訊息，BIOS 在一開始就已經燒錄到系統晶片當中，一般無法做修改。

BIOS 的作用就像是嵌入式系統中的 Boot loader 負責載入作業系統與對硬體做初始化的動作，不同的是 BIOS 除了載入作業系統，也會傳遞系統用的參數(中斷向量表等)，而 BootLoader 只是單純載入軟體。

BSP 是和作業系統綁在一起並一起在開發板上執行，即使在開機時 BIOS 與 BSP 的功能類似，但彼此間還是有明顯差異，BSP 本身還包含系統硬體相關的驅動，也可讓程式設計人員自行加入程式，甚至把上層開發軟體放入 BSP 中；而 BIOS 不能讓使用者修改，只能對參數作設定，也沒有硬體驅動程式。

比較物 項目	PC 上的 BIOS	BSP
功能	電腦開啟時的檢測 系統硬體的初始化 載入作業系統 讓作業系統對應體發出訊息	BootLoader 功能 系統硬體初始化 硬體驅動程式
修改性	燒入晶片後便不可修改，只可 設定參數	編譯前可修改， 也可自行加入應 用程式

表 2. PC 上的 BIOS 與 BSP 比較[1]

比較 WinCE、Linux 的 BSP

目前市面上的嵌入式作業系統成為百家爭鳴的局面，常見的作業系統有 Linux、WinCE、Vxworks、uClinux、PalmOS 等，由於每種作業系統彼此 BSP 有很大的差異，這節我們針對市占率較大的 WinCE 與 Linux 兩種作業系統的 BSP 來做比較。

比較物 項目	WinCE BSP	Linux BSP
商用系統	是	否
BSP 呈現方式	獨立出來 BSP	包含在 Kernel 原始碼 中
Boot loader	與 BSP 放在一起 規格較統一	自己獨立出來 版本多
結構	BSP 內容依資料夾分 類，結構清晰	包含在 Kernel 中，較不 好分別
硬體初始化程式碼	開機時與作業系統啟動 時可供用	開機時與作業系統啟動 時不可共用
核心空間與使用者 空間	在一起	分開
資料交換	較快	較慢
結構穩定性	較差	較好
作業系統原始碼	買版權才能取得	公開
相關應用多樣性	較少	較豐富
移植性	較方便	修改步驟較多

表 3. WinCE BSP 與 Linux BSP 比較[5]

WinCE 屬於商用型作業系統，所以需要購買版權才可得到系統原始碼，因此對於 WinCE 的相關應用軟體不像 Linux 廣泛；移植性方面，WinCE 為專門用於嵌入式系統的作業系統，所以會比由 x86 平台移植的 Linux 還容易，加上 WinCE 有獨立出來的 BSP，官方網頁也提出設計規範，修改 BSP 也比較容易，而 Linux 的 BSP 是包含在 Kernel 的原始碼中，除了要對目標平台硬體了解，對於 Linux kernel 運作方式也要熟悉；在 Boot Loader 方面，WinCE 的 Boot Loader 包含在 BSP 中，形式較統一，而 Linux 的 Boot Loader 與核心分開，某些開發板廠商會自行開發自己的 Boot Loader，而由於 Linux 的 Boot Loader 與核心分開，硬體初始化程式碼在開機與作業系統啟動時不能共同使用；最後討論到資料交換，Linux 核心空間與使用者空間分開，資料交換時較花時間，WinCE 兩個空間在一起，相對傳輸速度較快，不過 Linux 也因為這樣的架構使系統穩定性較好；表 3 為整理 WinCE BSP 與 Linux BSP 之比較。

4. Turn-Key 實作：含 BSP 與不含 BSP[5][6][7]

在這單元我們會編譯 AT2440EVB-II 開發板廠商提供調適過的 Linux Kernel 2.6.18.2 版本(已包含 BSP)與編譯 Linux Kernel 2.6.18.2(未包含 BSP)，移植(Porting)到開發板運行的情形，比較有無 BSP 對於開發板的硬體支援程度。

Kernel 2.6.18.2(有含 BSP)移植過程

移植一開始我們先將廠商提供的 Boot Loader(ATBoot)燒入開發板中，並在開發平台安裝交叉編譯環境(GCC 3.4.1)，由於廠商已經對核心做過調式，我們只需修改 Makefile 檔與 NAND Flash 配置便可直接進行編譯，將編譯好的映像檔燒入開發板，再燒入用 Busybox 製作的 File System 映像檔後，便可將 Linux 運行於開發板上。

Kernel 2.6.18.2(不含 BSP)移植過程

這裡我們先將 Linux Kernel 2.6.18.2 版本的原始碼下載下來，由於目標是移植核心，所以 Boot Loader 與 root file system 不會做任何修改，下載後的核心首先要修改核心根目錄下的 Makefile 檔，加入目標平台與交叉編譯的參數，再修改檔案/arch/arm/mach-s3c2410/common-smdk.c，配合開發板 Nand Flash 記憶體分區的方式來做修改，最後在根目錄執行 make menuconfig 指令，選擇系統類型、設定參數與加入需要的模組(驅動程式等)，我們參考廠商提供核心中.config 檔的設定方式，設定完後進行編譯，最後將編譯好的映像檔燒入開發板，便可順利執行 Linux 作業系統。

Linux 核心硬體支援度比較

	Turn-Key 含 BSP	Turn-Key 不含 BSP
(1)系統啟動 LED 燈	會亮	不會亮
(2)CPU、DRAM、FLASH	正常啟用	正常啟用
(3)網路功能	支援	不支援，須修改核心
(4)音樂輸出	支援	不支援，須修改核心
(5)USB 介面	支援	支援
(6)LCD 顯示	支援	不支援，須修改核心
(7)SD 卡讀寫	支援	不支援，須修改核心

表 4：硬體支援度比較表

這裡列出開發板所提供的硬體配件，來比較兩個 Turn-Key(Linux kernel 含 BSP + Root File System 與 Linux Kernel 不含 BSP + Root File System)對硬體支援度比較，比較標準以是否可控制該硬體為依據，結果如表 4。

在討論對硬體的支援程度時，我們對每個項目都在編譯前的 menuconfig 做過設定，把需要的模組加入我們的核心中，不過每塊開發板的硬體規格不大相同，所以加入的模組還是需要修改程式碼。如表 4，在(1)系統啟動 LED 燈的部分，Turn-Key 含 BSP 在核心啟動時會亮，而 Turn-Key 不含 BSP 核心啟動時不會反應；(2) CPU、DRAM、FLASH 方面，在之前提到，要使嵌入式系統順利運作，一定要具備 CPU、DRAM 與 FLASH，我們只需修改 FLASH 記憶體配置便可讓核心順利運行。

在開發板週邊硬體方面，(3)網路功能，AT2440EVB-II 使用的網路晶片為 Davicom DM9000AE，而在 menuconfig 選項中只包含 DM9000 選項，因此須另外加入驅動程式才可支援；(4)音樂輸出方面，音效驅動需要移植，(5)USB 介面方面，兩者皆支援 USB，我們分別插入 USB 鍵盤，可以馬上被核心偵測出來；(6)LCD 顯示方面，我們事先把 QT(一種跨平台的圖形介面軟體)執行檔複製到 root file system 中，Turn-Key 含 BSP 的部分，廠商有另外在 kernel 中加一個 SBZ2440.c 的檔案對 LCD 驅動作調整，因此可以針對 LCD 螢幕大小而調整，在核心啟動後將 QT 運行於系統中，而 Turn-Key 不含 BSP 的系統在 LCD 驅動沒做修改，所以無法支援；(7)最後在 SD 卡讀寫方面，Turn-Key 含 BSP 在系統啟動時便會偵測是否插入 SD 卡，開機後可以看到/dev/mmcblk0b1 這檔案，只要將此資料夾掛載便可看到 SD 卡的內容，而相對 Turn-Key 不含 BSP 的系統在開機後無此檔案出現。

5. 結論

在開發 BSP 直到整理成 Turn-Key 的過程中，我們常討論到底層驅動程式的實作，須對硬體有足夠的了解，才可以進行開發的作業，而因為現在驅動程式逐漸模組化，只要先了解規定的 I/O 接口再來實作便較容易上手。

隨著 Linux Kernel 不斷改版，很多硬體廠商願意提供 BSP 給 Linux Kernel

團隊做整合，我們只要把編譯好的 Linux kernel 燒入嵌入式系統便可以讓 Linux 運行，不過經由這次實驗可知，每塊開發板的週邊硬體有不同規格，核心所提供的驅動程式不一定適用，因此需要廠商提供 BSP 才能對硬體做完整的支援，不過在 Linux 核心中，已包含大部分硬體規格的驅動，我們再針對需求對檔案做修改，不用從零開始寫一份驅動程式，在未來的 Linux 核心版本，會有更多對硬體的支援，特別是較常見的硬體架構，如 x86、ARM、PXA2XX 系列等，在移植時只需按照標準程序便可順利運行。

在未來的嵌入式系統會有更多硬體規格，沒有 BSP 的 Linux Kernel 無法支援這些硬體，所以 BSP 需要該硬體製造商來提供，而廠商所提供針對該硬體的 BSP 只會是其中一個版本的 Linux Kernel，之後想對更新版本的核心作移植，建議參考廠商提供含 BSP 的 Linux Kernel 的調適方式，自行來完成移植的工作。

6. 參考資料

- [1] 曹樂淇、曹世強、林盈達, 嵌入式作業系統, 2003
http://speed.cis.nctu.edu.tw/~ydlin/miscpub/survey_embOS.pdf
- [2] BSP 概念解析, <http://www.embhelp.com/drew/mypage/bsp.htm>
- [3] Big Book of BSP, windowscebsp
<http://channel9.msdn.com/wiki/cedevolver/windowscebsp/>
- [4] Microsoft Windows CE 5.0 Board Support Package, Boot Loader, and Kernel Startup Sequence, msdn ,<http://msdn.microsoft.com/en-us/library/aa446905.aspx/>
- [5] JIANG Kunpeng, LU Dongxin, MIAO Jing, Comparison and Analysis of Development of Board Support Package between Linux and VxWorks, Institute of Chengdu, ZTE Corporation, Chengdu, 2003
- [6] 陳銘宏 林盈達, 移植 Linux 至 ARM 嵌入式系統
http://speed.cis.nctu.edu.tw/~ydlin/miscpub/hands-on_port-linux-to-ARM.pdf
- [7] 長高科技, ARM9 S3C2440 嵌入式系統實作(LINUX 及 MiniGUI 上層應用實驗篇), 長高科技圖書, 2007