```
<BOARD name="Creator" manufacture="Microtime">
    <SOC family="ARM" core="920T" manufacture="Samaung" id="S3C2410X01">
        <OCF name="core" defusr="&hardware;">
            <FUNC name="SetProcessorModeUsr" lang="ASM">
                mrsr0,cpsr
                bicr0,r0,#0x1f
                orrr0,r0,#0x10
                msrcpsr,r0
                movpc,lr
            </FUNC>
            ...
        <OCF name="timer" defusr="&hardware;">
            <FUNC name="Timer4Init" return="void" argu="void" lang="C">
                int timerloadvalue, tenms=100;
                timerloadvalue = 50*1000000/16/2/tenms;
                rTCFG0 = 0x0f00;
                rTCNTB4 = timerloadvalue;
                rTCON = 0x600000;
                rTCON = 0x500000;
            </FUNC>
            ...
<FUNC name="Timer0Init" return="void" argu="void" lang="C">
                ...
            </FUNC>
        </OCF>
        ...
    </SOC>
</BOARD>
```

第三圖

```
<BOOTSTRAP>
    <ENV_VAR>
        <FLASH_START>
            &flash_saddr;
        </FLASH_START>
        <FLASH_END>
            &flash_saddr;+&flash_size;-1
        </FLASH_END>
        <SDRAM_START>
            &sdram_saddr;
        </SDRAM_START>
        <TEXT_BASE>
            &sdram_saddr;+0x1f00000
        </TEXT_BASE>
        ...
    </ENV_VAR>
    <INIT_HEAD>
        <BL fname="SetProcessorModeSvc"></BL>
        <BL fname="WatchdogOff"></BL>
        <BL fname="InterruptAllServiceOff"></BL>
        <BL fname="SubInterruptAllServiceOff"></BL>
        <BL fname="SetInterruptModeIrq"></BL>
        <BL fname="ClearAllInterruptPending"></BL>
        <BL fname="SetMemoryControl"></BL>
        <BL fname="ClockDividerControl1_2_4"></BL>
        <BL fname="SetPLL"></BL>
        <BL fname="MMU_SetAsyncBusMode"></BL>
        ...
    </INIT_HEAD>
    <INIT_CMAIN>
        <INVOKE fname="SerialInit"></INVOKE>
        <INVOKE fname="Timer4Init"></INVOKE>
        <INVOKE fname="SetIOport"></INVOKE>
        <INVOKE fname="EnableInterrupts"> </INVOKE>
        ...
    </INIT_CMAIN>
</BOOTSTRAP>
```

第四圖

```
<!ELEMENT BOARD (SOC, PERIDEV, RELATEDTOOL, BOOTSTRAP)>
<!ATTLIST BOARD
name CDATA #REQUIRED
manufacturer CDATA #REQUIRED

<!ELEMENT SOC (OCF*)>
<!ATTLIST SOC
family CDATA #REQUIRED
core CDATA #REQUIRED
manufacturer CDATA #REQUIRED
id CDATA #REQUIRED>
    <!ELEMENT OCF (FUNC*)>
    <!ATTLIST OCF
    name CDATA #REQUIRED
    defusr CDATA #REQUIRED
        <!ELEMENT FUNC (#PCDATA)>
        <!ATTLIST FUNC
        name CDATA #REQUIRED
        return CDATA #IMPLIED
        argu CDATA #IMPLIED
        lang (ASM|MACRO|C) #REQUIRED>

<!ELEMENT PERIDEV (DEV*)>
...
  <!ELEMENT TOOL (#PCDATA)>
  ...
  <!ELEMENT BOOTSTRAP (ENV_VAR, INIT_HEAD, INIT_CMAIN)>
  ...
```

第五圖

I294589

公告本

修正本

發明專利說明書

95年6月9日 修正補充

（本說明書格式、順序及粗體字，請勿任意更動，※記號部分請勿填寫）

※ 申請案號：94146f.6

※ 申請日期：94.12.27 ※IPC 分類：G06F9/45、17/50 (2006.01)

一、發明名稱：（中文/英文）

應用於嵌入式系統輔助移植之通用 BSP 工具及其應用方法

二、申請人：（共 1 人）

姓名或名稱：（中文/英文）

國立中正大學

代表人：（中文/英文）羅仁權

住居所或營業所地址：（中文/英文）

嘉義縣民雄鄉大學路 168 號

國　籍：（中文/英文）中華民國/TW

三、發明人：（共 4 人）

姓　名：（中文/英文）

羅習五/N123158798　　林信宏/E122653028

薛智文/D120509375

周雍傑/U121189586

國　籍：（中文/英文）

中華民國/TW

四、聲明事項：

☑ 主張專利法第二十二條第二項☑第一款或☐第二款規定之事實，其事實發生日期為：2005 年 06 月 27 日。

☐ 申請前已向下列國家（地區）申請專利：

【格式請依：受理國家（地區）、申請日、申請案號　順序註記】

☐ 有主張專利法第二十七條第一項國際優先權：

☐ 無主張專利法第二十七條第一項國際優先權：

☐ 主張專利法第二十九條第一項國內優先權：

【格式請依：申請日、申請案號　順序註記】

☐ 主張專利法第三十條生物材料：

☐ 須寄存生物材料者：

國內生物材料 【格式請依：寄存機構、日期、號碼　順序註記】

國外生物材料 【格式請依：寄存國家、機構、日期、號碼　順序註記】

☐ 不須寄存生物材料者：

所屬技術領域中具有通常知識者易於獲得時，不須寄存。

九、發明說明：

【發明所屬之技術領域】

本發明係有關一種主機板支援套件（Board Support Package, BSP）工具，特別是指一種應用於嵌入式系統輔助移植之通用 BSP 工具及其應用方法。

【先前技術】

隨著電腦的系統效能與運算能力不斷提升，嵌入式作業系統在嵌入式系統中扮演了很重要的一環，良好的作業系統可有效的利用系統資源並控制執行要素。一般來說，移植是使程式可在不同硬體環境下執行的必要程序，系統開發員為了能執行嵌入式系統，通常會移植作業系統或應用程式到嵌入式系統中，此時會需要更動作業系統或底層程式碼。

主機板支援套件（Board Support Package, BSP），是介於主機板和作業系統之間的一層，屬於作業系統的一部分，主要目的是為了支持作業系統，使之能夠運行於主機板上。BSP 是相對於硬體平台與作業系統而言的，不同的作業系統對應於不同定義形式的 BSP，例如 VxWorks 和 Linux 系統的 BSP 相對於某一處理器（Processor）來說儘管實現的功能一樣，可是寫法和接口定義可能完全不同。同樣地，不同的硬體平台也定義了不同形式的 BSP，例如 Linux 在不同處理器的 BSP 也可能不同。由於 BSP 的程式化過程大多數是在某一個成型的 BSP 模板上進行修改，故編寫 BSP 必須按照該系統 BSP 的定義形式來寫，這樣才能與上層作業系統保持正確的接口，良好支持上層作業系統。

目前市場上多種結構的嵌入式處理器並存，如 ARM、MIPS、XScale 等處理器，為了性能的需要，週邊設備也會有不同的選擇和定義。一個嵌入式操作系統針對不同的處理器會有不同的 BSP，即使是同一種處理器，由於外設的一點差別，如外部擴展動態隨機存取記憶體的大小或類型改變，BSP 相應的部分也不一樣，所以必須根據硬體規格來設計編寫及修改 BSP，以確保系統的正常運行。

在先前技術中，開發員要建立一個嵌入式系統，通常會使用包含

廠商所提供之硬體平台、作業系統及工具組等的全方位解決方案，而開發員更需要因應嵌入式系統中各種環境需求來修改作業系統，如即時控制（real-time control）或較小的佔用空間（footprint）。若作業系統不能滿足所有的環境需求，開發員就必須移植其他的作業系統到此硬體平台上來執行，然而，要在新的硬體平台上運作，開發員亦須移植嵌入式應用程式，在移植作業系統及應用程式到新的硬體平台上之前，開發員首先必須了解硬體平台的特徵，但硬體廠商有時並不會主動提供相關細節，故開發員必須先針對低階程式碼進行了解，如開機程式或裝置驅動程式，此部份需花費相當的時間與精力。而先前技術中並沒有可自動及系統化的方法解決移植的問題，縮短移植程序可使嵌入式系統上市的準備時間縮短。

因此，本發明即針對上述習知技術之數項缺失，提出一種應用於嵌入式系統輔助移植之通用 BSP 工具及其應用方法，以有效克服上述之該等問題。

【發明內容】

本發明之主要目的在提供一種應用於嵌入式系統輔助移植之通用BSP工具及其應用方法，其係可降低移植作業系統或應用程式到嵌入式硬體平台時的障礙，使市場需求多元且成長快速。

本發明之另一目的在提供一種應用於嵌入式系統輔助移植之通用BSP工具及其應用方法，其係提供硬體製造商一種普通且明確的標準，以描述嵌入式硬體的規格特性與作業系統的開機程序。

本發明之再一目的在提供一種應用於嵌入式系統輔助移植之通用BSP工具及其應用方法，其係可分析BSPXML文件中所描述之符合該嵌入式平台硬體規格的作業系統開機程序，並提供一低階硬體控制函式，可讓作業系統或應用程式在開機後直接控制硬體。

為達上述之目的，本發明提供一種應用於嵌入式系統輔助移植之通用 BSP 工具及其應用方法，其係利用一 BSP 語法分析器分析一符合 XML 規則之 BSPXML 文件，以獲得一符合嵌入式平台之硬體規格及作業系統或應用程式之開機程序，其中 BSPXML 文件更可藉由定義一 BSP 文

檔類型定義（Document Type Definition, DTD）來擴充延伸；再利用一程式碼產生器接收 BSP 語法分析程式所產生之硬體規格與開機程序等資訊後，產生嵌入式平台之開機程式及低階硬體控制函式，以供作業系統或應用程式使用。

底下藉由具體實施例詳加說明，當更容易瞭解本發明之目的、技術內容、特點及其所達成之功效。

【實施方式】

本發明係提供一種應用於嵌入式系統輔助移植之通用 BSP 工具，其係針對嵌入式硬體平台的規格及作業系統開機程序來設計，並將規格及程序撰寫於一開放式且可重複使用的BSPXML文件中，本發明之BSP工具係可從語法上分析BSPXML文件並產生目標平台上內嵌之作業系統或應用程式的開機程式。

如第一圖所示，本發明所提供之通用 BSP 工具包括一 BSP 語法分析器 32 及一程式碼產生器 34，BSP 語法分析器 32 係用以分析 BSPXML 文件 12、22，此 BSPXML 文件 12、22 分別描述所屬嵌入式平台之硬體規格及作業系統或應用程式之開機程序，其中 BSPXML 文件 12 描述硬體製造商所提供之硬體架構 10，而 BSPXML 文件 22 則描述新硬體平台 20 的硬體規格；而程式碼產生器 34 則在接收 BSP 語法分析器 32 所產生之硬體規格與開機程序等資訊後，產生嵌入式平台之開機程式及低階硬體控制函式，以供作業系統或應用程式使用。

BSPXML 文件 12、22 符合 XML 格式並具有開放及可重複使用的特性，其中包含有四個元素：系統單晶片 122、222、週邊設備 124、224、工具組 126、226 及開機程序 128、228；在移植作業系統至不同硬體平台的時候，如硬體平台 10、20，需要使用符合該硬體平台的系統單晶片 122、222 及週邊設備 124、224，因此系統單晶片 122、222 及週邊設備 124、224 便依此而設定。在 BSPXML 文件 12、22 中的工具組 126、226 中更可包括作業系統或應用程式所需要之編譯器等工具組與 makefile 檔等。

系統單晶片 122、222 用以定義嵌入式系統處理器之硬體規格及相

關程式碼，包括處理器模式、中斷模式及遮罩，以及監時器（watchdog timer）等，硬體製造商亦可利用系統單晶片 122、222 來描述記憶體控制器、處理單元及快取記憶體等硬體裝置之規格，甚至啟用與停用的各控制器程式碼亦可撰寫於系統單晶片 122、222 中；週邊設備 124、224 用以敘述嵌入式系統中週邊裝置，包括有通用非同步收發傳輸器（Universal Asynchronous Receiver/Transmitter, UART）、快閃記憶體及乙太網路控制器（Ethernet controller）等裝置。硬體製造商會提供基本的驅動程式，在嵌入式平台中每一硬體平台之記憶映象皆需詳細地描述。

而工具組 126、226 即用於描述每一硬體平台所需要特定之工具組，利用交叉編譯器（cross compiler）、Linux 系統最底層之應用程式開發介面 GlibC、除錯器及函式庫等低階開發工具，以產生開機程式。而開機程序 128、228 則描述作業系統或應用程式開機時的流程，如將硬體初始化，可使系統開發員 30 透過程式碼產生器 34 產生適當的開機程式，並將作業系統或應用程式載入記憶體中。

BSP 文檔類型定義（Document Type Definition, DTD）36 在本發明中係利用定義一合格元素列表（list of legal elements）來定義 BSPXML 文件 12、22 之規則架構，BSP 文檔類型定義 36 係通過元素宣告（element declaration）及屬性列表宣告（attribute-list declaration）來定義，其中元素宣告係用以給 BSPXML 文件 12、22 中有效之元素集合命名，並詳細說明每一元素中包含的特性資料，屬性列表宣告則為每一宣告元素的有效屬性集合命名，包括每一屬性值（attribute value）的類型。透過此 BSP 文檔類型定義 36，硬體製造商所提供之硬體架構 10 及系統開發員 30 可互換資料。一開始可先提供一標準化的 BSP 文檔類型定義 34，硬體製造商並提供其嵌入式系統之 BSPXML 文件 12，當硬體製造商新增或修改硬體架構時，則可新增新的元素或屬性到 BSP 文檔類型定義 34 中，以延展擴充 BSPXML 文件 12 之內容。

因此，當出現新的硬體平台 20 需要進行移植（porting）時，硬

體製造商可依照此新硬體平台 20 之硬體規格提供一 BSPXML 文件 22，而系統開發員即可依據此 BSPXML 文件 22 寫出或重複利用所要移植的作業系統或應用程式的開機程序 228。BSP 語法分析器 32 將欲移植之目標新硬體平台 20 的 BSPXML 文件 22 分析過後，由程式碼產生器 34 利用 BSPXML 文件 22 之程式語言來自動產生一開機程式碼，系統開發員 30 只需利用此開機程式碼編譯並連結至作業系統或應用程式，即可結束移植的動作，啟動新硬體平台 20。

若系統開發員 30 欲利用本發明將一個新的作業系統、應用程式或從未運用本發明來移植的作業系統或應用程式移植到新硬體平台 20，首先系統開發員 30 需要新硬體平台 20 之 BSPXML 文件 22，此時硬體製造商僅需提供 BSPXML 中描述新硬體平台 20 的系統單晶片 222、週邊設備 224 及工具組 226 的部分，系統開發員 30 在第一次移植某作業系統時需要花時間撰寫 BSPXML 文件來描述其開機程序 228。往後若要移植不同作業系統到此新硬體平台 20 時，則可使用先前所撰寫之 BSPXML 文件有關硬體平台描述的部分 222、224、226；當該作業系統或應用程式已完成移植後，其 BSPXML 文件的開機程序描述 228 可再次利用於不同硬體平台的移植，達到重複使用的目的，免除先前技術中系統開發員在面對不同系統之間的移植時，需要了解多種硬體平台與作業系統或應用程式架構的困難，利用本發明，系統開發員 30 可不費力的移植一普通的作業系統或應用程式到一嵌入式平台上。

請同時參考第二圖所示，其為本發明可提供之低階函式 40，在作業系統核心（OS kernel）中利用此些低階函式 40 以控制新硬體平台 20，如計時器初始化、啟用與停用記憶體控制器及中斷等，這些低階函式 40 更可在開機後運用於作業系統及應用程式中，系統開發員 30 利用這些低階函式 40 的幫助，可輕易移植低階硬體控制函式以簡化作業系統低階函式移植的程序。

第三圖為一 BSPXML 文件實施例之部分程式碼，其中詳述了 Microtime Inc. 所製造主機板之硬體規格，其中與系統單晶片元素處理程序有關之處理器模式、中斷模式與遮罩及監時器(watchdog timer)

等設定皆描述於系統單晶片中;BSPXML 文件中 OCF(On Chip Functions)元素係用於定義晶片上的控制函式，如 SetProcessorModeUsr 函式定義轉換處理器到使用者模式（user mode），TimerInit 相關函式用以將晶片上之計時器初始化。硬體製造商需提供以組合語言或 C 語言所撰寫之 OCF 函式。

第四圖為撰寫於 BSPXML 文件中作業系統的開機程序，使 BSP 工具可產生適當的開機程式碼。其中 ENV_VAR 中所包含之程式碼係定義作業系統中之環境變數，如記憶體規劃，INIT_HEAD 及 INIT_CMAIN 中之程式碼則定義了利用組合語言及 C 語言有順序的描述作業系統開機流程兩個部分。利用這些函式，作業系統不需要很詳細的硬體規格細節即可對新硬體平台進行初始化，使移植更容易。在第四圖之實施例中，INIT_HEAD 函式先將處理程序轉為 svc 模式，再將監時器及中斷等命令停止;接著執行的 INIT_CMAIN 函式則繼續初始化該新硬體平台，最後，在結束所有開機程序的執行後，開機程式碼返回由作業系統控制。

BSPXML 是由 BSP 文檔類型定義來界定規則，第五圖為 BSP 文檔類型定義之實施例，敘述了 BSP 文檔類型定義中系統單晶片元素及其子元素 OCF 及 FUNC。必要的屬性皆被清楚的定義在 BSP 文檔類型定義中，使硬體製造商、系統開發員及作業系統的程式員皆可瞭解 BSPXML 的語法從而使用 BSPXML 文件。

唯以上所述者，僅為本發明之較佳實施例而已，並非用來限定本發明實施之範圍。故即凡依本發明申請範圍所述之特徵及精神所為之均等變化或修飾，均應包括於本發明之申請專利範圍內。

【圖式簡單說明】

第一圖為本發明所提供之通用 BSP 工具與硬體平台之示意圖。

第二圖為本發明可提供之低階函式示意圖。

第三圖為本發明中系統單晶片元素之部分 BSPXML 文件之實施例。

第四圖為本發明中開機程序元素之部分 BSPXML 文件之實施例。

第五圖為本發明中部份 BSP 文檔類型定義之實施例。

【主要元件符號說明】

10 硬體製造商所提供之硬體架構

12 BSPXML 文件

122 系統單晶片（System on chip, SoC）

124 週邊設備

126 工具組

128 開機程序

20 新硬體平台

22 BSPXML 文件

222 系統單晶片（System on chip, SoC）

224 週邊設備

226 工具組

228 開機程序

30 系統開發員

32 BSP 語法分析器

34 程式碼產生器

36 BSP 文檔類型定義

38 輸出端

40 低階函式

五、中文發明摘要：

本發明提供一種應用於嵌入式系統輔助移植之通用 BSP 工具及其方法，其係包括一 BSP 語法分析器及一程式產生器，用以分析一符合 XML 格式並具有開放及可複寫特性之 BSPXML 文件，此 BSPXML 文件中描述嵌入式平台之硬體規格及作業系統之開機程序，而本發明在分析 BSPXML 文件後可自動產生指定嵌入式平台上的開機程式，以供指定之作業系統或應用程式使用。且本發明所提供之低階硬體控制函式可讓作業系統或應用程式在開機程序後直接控制硬體，簡化與自動化移植（porting）流程，達到縮短嵌入式系統開發時間以提早上市之目的。

六、英文發明摘要：

十、申請專利範圍：

1. 一種應用於嵌入式系統輔助移植之通用 BSP 工具，包括：

一 BSP 語法分析器，用以分析一 BSPXML 文件，以獲得一嵌入式平台之硬體規格及作業系統或應用程式之開機程序；以及

一程式碼產生器，其接收該 BSP 語法分析程式所產生之硬體規格與開機程序等資訊後，產生該嵌入式平台之開機程式及低階硬體控制函式，以供該作業系統或應用程式使用。

2. 如申請專利範圍第 1 項所述之應用於嵌入式系統輔助移植之通用 BSP 工具，其中該 BSPXML 文件係使用 XML 格式撰寫，具有開放及可重覆使用的特性。

3. 如申請專利範圍第 1 項所述之應用於嵌入式系統輔助移植之通用 BSP 工具，其中該 BSPXML 文件可藉由定義一 BSP 文檔類型定義（Document Type Definition, DTD）來擴充延伸，該 BSP 文檔類型定義係定義 BSPXML 規則來描述該嵌入式平台之硬體規格及該作業系統或應用程式之開機程序。

4. 如申請專利範圍第 1 項所述之應用於嵌入式系統輔助移植之通用 BSP 工具，其中該 BSPXML 文件中所描述該嵌入式平台之硬體規格係包括可用之暫存器、記憶映像（memory mapping）、控制該嵌入式平台之程式碼及用以開發該嵌入式平台之工具組。

5. 如申請專利範圍第 4 項所述之應用於嵌入式系統輔助移植之通用 BSP 工具，其中該可用之暫存器包括暫存器名稱、映射記憶體位址（mapped memory address）、暫存器位元定義（register bit definition）等。

6. 如申請專利範圍第 4 項所述之應用於嵌入式系統輔助移植之通用 BSP 工具，其中該記憶映像（memory mapping）係包括暫存器、ROM、RAM、快閃記憶體及其他裝置於該硬體平台上對映之記憶體位址。

7. 如申請專利範圍第 4 項所述之應用於嵌入式系統輔助移植之通用 BSP 工具，其中該工具組包括編譯器、除錯器、函式庫等所需工具。

8. 如申請專利範圍第 1 項所述之應用於嵌入式系統輔助移植之通用 BSP 工具，其中該 BSPXML 文件中之該開機程序係包括硬體初始化、將作業系統

或應用程式載入記憶體等步驟。

9. 如申請專利範圍第 1 項所述之應用於嵌入式系統輔助移植之通用 BSP 工具，其中該 BSPXML 文件中更可包括該作業系統或應用軟體之 makefile 檔案。

10. 如申請專利範圍第 1 項所述之應用於嵌入式系統輔助移植之通用 BSP 工具，其中該程式碼產生器係利用該 BSPXML 文件之描述來自動使用程式語言產生一開機程式碼，用以編譯並連結至該作業系統或應用程式，以啟動該嵌入式平台。

11. 如申請專利範圍第 1 項所述之應用於嵌入式系統輔助移植之通用 BSP 工具，其中該低階硬體控制函式係利用程式語言所撰寫之複數程式碼直接控制該嵌入式平台，包括存取暫存器之值、記憶體位置及開機後作業系統或應用程式所呼叫控制硬體之程式碼。

12. 如申請專利範圍第 10 項或第 11 項所述之應用於嵌入式系統輔助移植之通用 BSP 工具，其中該程式語言可為組合語言、C 語言或其它可支援該嵌入式平台之程式語言。

13. 如申請專利範圍第 1 項所述之應用於嵌入式系統輔助移植之通用 BSP 工具，其中該 BSP 文檔類型定義中之 BSPXML 文件係通過元素宣告(element declaration)及屬性列表宣告(attribute-list declaration)來定義。

14. 如申請專利範圍第 13 項所述之應用於嵌入式系統輔助移植之通用 BSP 工具，其中該元素宣告係用以給該 BSPXML 文件中有效之元素集合命名，並詳細說明每一元素中包含的特性資料。

15. 如申請專利範圍第 13 項所述之應用於嵌入式系統輔助移植之通用 BSP 工具，其中該屬性列表宣告為每一宣告元素的有效屬性集合命名，包括每一屬性值 (attribute value) 的類型。

16. 一種應用嵌入式系統輔助移植 BSP 工具之方法，包括以下步驟：

提供一 BSPXML 文件，其中係分別描述有欲移植之目標平台的硬體規格及欲移植之作業系統的開機流程；

利用一 BSP 語法分析器分析該目標平台之 BSPXML 文件，由一程式碼產生器利用程式語言來自動產生一開機程式碼；以及

利用該開機程式碼編譯並連結至作業系統或應用程式。

17. 如申請專利範圍第 16 項所述之應用嵌入式系統輔助移植 BSP 工具之方法，其中該 BSPXML 文件係使用 XML 格式撰寫，具有開放及可重覆使用的特性。

18. 如申請專利範圍第 16 項所述之應用嵌入式系統輔助移植 BSP 工具之方法，其中該 BSPXML 文件可藉由定義一 BSP 文檔類型定義(Document Type Definition, DTD) 來擴充延伸，該 BSP 文檔類型定義係定義 BSPXML 規則來描述該嵌入式平台之硬體規格及該作業系統或應用程式之開機程序。

19. 如申請專利範圍第 16 項所述之應用嵌入式系統輔助移植 BSP 工具之方法，其中該程式語言可為組合語言、C 語言或其它可支援該嵌入式平台之程式語言。

20. 如申請專利範圍第 16 項所述之應用嵌入式系統輔助移植 BSP 工具之方法，其中該 BSP 文檔類型定義中之 BSPXML 文件係通過元素宣告(element declaration)及屬性列表宣告(attribute-list declaration)來定義。

21. 如申請專利範圍第 20 項所述之應用嵌入式系統輔助移植 BSP 工具之方法，其中該元素宣告係用以給該 BSPXML 文件中有效之元素集合命名，並詳細說明每一元素中包含的特性資料。

22. 如申請專利範圍第 20 項所述之應用嵌入式系統輔助移植 BSP 工具之方法，其中該屬性列表宣告為每一宣告元素的有效屬性集合命名，包括每一屬性值 (attribute value) 的類型。

七、指定代表圖：

(一)本案指定代表圖為：第（ 一 ）圖。

(二)本代表圖之元件符號簡單說明:

10 硬體製造商所提供之硬體架構

12 BSPXML 文件	122 系統單晶片
124 週邊設備	126 工具組
128 開機程序	20 新硬體平台
22 BSPXML 文件	222 系統單晶片
224 週邊設備	226 工具組
228 開機程序	30 系統開發員
32 BSP 語法分析器	34 程式碼產生器
36 BSP 文檔類型定義	38 輸出端

八、本案若有化學式時，請揭示最能顯示發明特徵的化學式：