

國立交通大學

管理學院科技法律學程

碩士論文

台灣廠商使用開放原始碼的法律問題研究
- 以使用 GPLv2 授權條款的 Embedded Linux 為例

Legal Issues for Taiwanese Manufactures Using Open
Source Software

— Focusing on Embedded Linux under GPLv2

研究生：翁儷娟

指導教授：王敏銓 老師、林三元 老師

中華民國一〇一年六月

台灣廠商使用開放原始碼的法律問題研究
-以使用 GPLv2 授權條款的 Embedded Linux 為例

研究生：翁儷娟

Student : Weng, Li-Chuan

指導教授：王敏銓

Advisor : Prof. Wang, Min-Chiuan

林三元

Prof. Lin, San-Yuan



June 2011

Hsinchu, Taiwan, Republic of China

中華民國一〇一年六月

台灣廠商使用開放原始碼的法律問題研究

-以使用 GPLv2 授權條款的 Embedded Linux 為例

學生：翁儷娟

指導教授：王敏銓 老師

林三元 老師

國立交通大學管理學院科技法律學程

摘要

開放原始碼經常被用來加速嵌入式產品的開發時程，但伴隨開放原始碼而來的開放原始碼授權條款，對於商業使用卻帶有對立甚至制衡的色彩。本研究的重點在探究台灣的嵌入式產品開發廠商，在使用開放原始碼輔助產品開發時的法律問題，並以 Embedded Linux 作業系統的使用為討論對象。在嵌入式產品中的系統程式中，作業系統扮演著承上啓下的角色，對整體的系統程式具有相當大的影響。然而，在嵌入式產品中位居主流地位的 Linux 作業系統所選用的開放原始碼授權條款，又是以保障自由出名的通用公共授權條款第二版 (GNU General Public License Version 2, 以下簡稱為 GPLv2 授權條款)，因此廠商採用時莫不謹言慎行，擔心己身對於開放原始碼的不當使用而引發爭訟。本研究首先探究開放原始碼授權條款的內涵與法律定位、對相關判決與爭議來進行評析，接著對嵌入式產品開發過程中的晶片設計與系統開發兩個環節之相關開發者進行深度訪談，藉以瞭解當前台灣的嵌入式產品開發廠商使用開放原始碼時的作法，與可能伴隨的法律風險，本研究並進一步提供幾點建議作為相關廠商利用開放原始碼進行產品開發時參酌。此外，由於嵌入式產品的本質與個人電腦及工作站的本質有許多歧異，因此針對嵌入式產品的應用，GPLv2 授權條款的規範似應針對該等歧異而有不同考量，為此，本研究亦對 GPLv2 授權條款的規範內容提出修改的建議，期能找出兼顧廠商利益與社群理想的授權方式。

關鍵字：開放原始碼、嵌入式產品、授權條款、Linux、GPLv2

Legal Issues for Taiwanese Manufactures Using Open Source Software

— Focusing on Embedded Linux under GPLv2

student : Weng, Li-Chuan

Advisors : Prof. Wang, Min-Chiuan
Prof. Lin, San-Yuan

Institute of Technology Law
National Chiao Tung University

Abstract

Open source software is widely used to accelerate the schedule of embedded product development. However, the licenses used with the open source programs are considered to be opposite to business usage. The focus of the article is to discuss the legal issues that Taiwanese embedded product manufactures may encounter once they use open source software to assist the development of their products, using Linux operating system as an example. For embedded products, the operation system plays a key role and is very important for the system programs. However, the main stream operating system in embedded product markets, Linux, uses the GNU General Public License Version 2(GPLv2) that is famous of its demands for software freedom. Therefore, embedded product manufactures are very cautious once they choose to use Linux OS in their embedded products, in order to avoid litigation that might be caused by the misuse of open source software. This article first discusses the contents and legal issues of open source licenses, followed by analyzing related debates and cases. Afterwards, the author hold in-depth interviews with system engineers working in both the chip design house and system manufactures, in order to understand how Taiwanese embedded product manufactures act and the ensuing risks. We further provide several suggestions that embedded product manufactures may follow once they use open source software for their product development. Besides, due to the distinct nature of the embedded products and personal computers, it seems to us that GPLv2 should consider differently for applications of embedded products. Therefore, the author proposes a few suggestions for the GPLv2 and hope to find out a solution to accommodate both the benefits to manufactures and the spirit of open source software.

Keywords: open source, embedded product, license, Linux, GPLv2

誌謝

漫長的半工半讀生涯，終於到了劃下句點的時候，被大家不斷詢問何時可以畢業的日子也終於可以結束了。最初在工研院主管的推薦下，憑著一股衝動報考了交大管院專班的科技法律學程（科法所），很意外也很有榮幸的順利錄取。儘管在獲知錄取時，已經轉職到私人企業，但有幸遇到開明的主管，也因此開始了緊湊的在職進修生活。五年來，工作環境雖然因為各種內外因素的變動而不斷地更迭，但在科法所的學習卻未曾間斷。

本論文的完成，必須感謝論文指導老師王敏銓老師與林三元法官的熱心指導，老師們提供了許多專業而寶貴的意見，讓我在撰寫論文的過程中獲益良多。此外，口試委員林昱梅老師所提供的建議也相當寶貴，感謝三位老師對於口試本的認真批改，讓這份論文能更臻完善與周延。科法所的老師們，犧牲寶貴的週末時間教導我們，這些學習的過往都點滴在心頭。也謝謝本論文受訪者熱心的經驗分享，讓我的論文得以掌握第一手資料並忠實的反應目前實務界的作法。

儘管在報考科法所時，係以個人興趣為出發點，但是就讀以來的潛移默化，無形中對個人的職涯規劃也產生很大的影響。感謝這五年中在其樂達、聯詠、碩維與目前公司的主管、同事們的包容與支持，讓我得以在兼顧工作與學業的情況下，以緩慢但尚稱堅定的步伐取得科法所的學位。再者，能夠順利修完學分、完成此份論文，必須非常謝謝在科法所就讀時一起努力奮鬥、彼此打氣同學們，特別是同樣需補修許多先修課程而苦哈哈的科技組同學們。此外，也相當謝謝科法所的諸位助理們，在我們弄不清楚眾多流程與規定時，耐心的協助我們。

非常感謝我的家人，在我任性的選擇自己人生的道路時，相信我的判斷，也一直作為我的強力後盾。也要感謝身旁許多的朋友、各階段的師長與同學們，讓我知道儘管自己經常在作一些相當特立獨行的決定，但周圍的人還是支持我的。

受限於篇幅，無法逐一對周遭幫忙的人士表達謝意，在此感謝大家！

翁儷娟 2011/06/06（端午節）

目錄

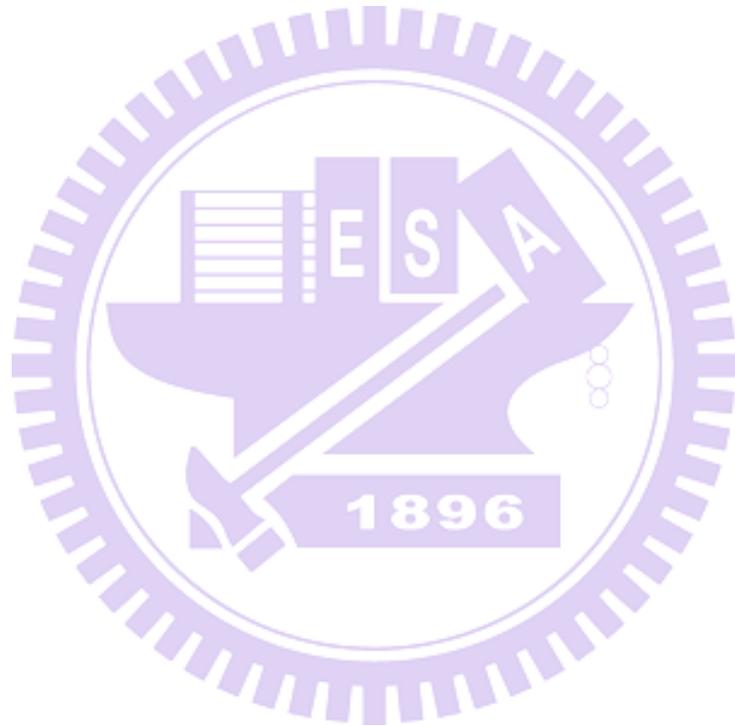
	頁次
摘要.....	i
Abstract.....	ii
誌謝.....	iii
目錄.....	iv
圖目錄.....	vii
表目錄.....	viii
第一章 緒論.....	1
1.1 研究動機.....	2
1.2 研究目的.....	3
1.3 論文架構.....	5
第二章 研究限制.....	7
2.1. 研究範圍的限制.....	7
2.1.1. 選擇 Embedded Linux 作為討論對象.....	7
2.1.2. 開放原始碼授權條款的選擇.....	8
2.1.3. 受訪對象的選擇.....	8
2.2. 資料蒐集的限制.....	8
2.3. 研究方法的限制.....	9
2.3.1. 文獻分析法的限制.....	9
2.3.2. 半結構式訪談的限制.....	9
第三章 技術背景.....	11
3.1 嵌入式產品.....	11
3.1.1. 嵌入式產品的硬體架構與產業分工.....	11
3.1.2. 嵌入式產品的軟體架構與開發環境.....	12
3.1.3. 嵌入式產品的作業系統.....	13
3.2 軟體的分類.....	14
3.3 嵌入式產品與個人電腦的差別.....	16
3.3.1. 使用場合與數量.....	17
3.3.2. 軟體與硬體的感受程度.....	17
3.3.3. 廠商的獲利來源.....	17
3.3.4. 產品的相容性.....	18
3.3.5. 系統軟體開發模式.....	19
3.4 開放原始碼與嵌入式產品.....	19

3.5	嵌入式產品中的 Linux 作業系統-Embedded Linux.....	20
第四章	開放原始碼授權條款.....	23
4.1	背景.....	23
4.1.1	開放原始碼授權條款的起源.....	23
4.1.2	開放原始碼授權條款與軟體.....	24
4.1.3	開放原始碼授權條款與網路架構.....	25
4.2	基本分類.....	26
4.2.1	互惠型開放原始碼授權條款.....	27
4.2.2	折衷型開放原始碼授權條款.....	28
4.2.3	學術型開放原始碼授權條款.....	29
4.2.4	相容性.....	29
4.3	法律定位.....	32
4.3.1	開放原始碼授權條款與著作權.....	33
4.3.2	開放原始碼授權條款與專利.....	34
4.3.3	開放原始碼授權條款與競爭法.....	36
4.4	小結.....	37
第五章	Embedded Linux 的授權範圍.....	39
5.1	Embedded Linux 與 GPLv2 授權條款.....	39
5.2	GPLv2 授權條款的規範.....	40
5.2.1	GPLv2 前言.....	41
5.2.2	GPLv2 第 5 條.....	42
5.3	GPLv2 授權條款中與嵌入式產品特質相關的條文.....	42
5.3.1	GPLv2 第 0 條.....	43
5.3.2	GPLv2 第 1 條.....	44
5.3.3	GPLv2 第 2 條.....	46
5.3.4	GPLv2 第 3 條.....	49
5.3.5	GPLv2 第 4 條.....	50
5.3.6	GPLv2 第 6 條.....	51
5.3.7	小結.....	52
5.4	非與嵌入式產品特性相關的條款.....	53
5.5	Linux 的自我限制或棄權.....	53
5.6	小結.....	55
第六章	使用開放原始碼的嵌入式產品-爭議、衝突與作法.....	57
6.1	遵守 GPLv2 授權條款規定之作法.....	57
6.1.1	提供程式的原始碼但用科技保護措施規避.....	58
6.1.2	分開散布.....	59
6.2	違反開放原始碼授權條款所衍生的嵌入式產品訴訟.....	61
6.2.1	違反 GPLv2 所衍生的訴訟.....	62

6.2.2	違反他種開放原始碼授權條款所衍生的訴訟	66
6.3	使用 Embedded Linux 並遵守條款規定的作法	68
6.3.1	透過提供 Embedded Linux 的技術服務而獲利	69
6.3.2	以 Linux 為基礎的各種開發平台.....	69
6.4	小結	72
第七章	實證研究分析-深度訪談.....	75
7.1	研究設計	75
7.1.1	研究假設	76
7.1.2	受訪對象	78
7.2	訪談內容與歸納	79
7.2.1	原始碼的取得來源、管理方式與提供方式	79
7.2.2	作業系統的使用與否及其程式來源	84
7.2.3	開放原始碼之於嵌入式產品的系統程式開發	86
7.2.4	系統程式的原始碼管理與軟體授權相關的法律問題	90
7.2.5	使用開放原始碼的法律爭議	92
7.3	小結	94
第八章	建議、結論與未來研究方向.....	97
8.1	建議與結論	97
8.1.1	對 GPLv2 授權條款的修改建議.....	97
8.1.2	根據現行 GPLv2 授權條款給予廠商的建議.....	101
8.1.3	結論	107
8.2	未來研究建議	109
附錄一	訪談資料.....	117
附錄一	自由軟體使用判斷流程圖.....	120
附錄三	GPLv2 授權條款.....	121
簡歷	127

圖目錄

	頁次
圖 1，本研究主題與嵌入式產品、開放原始碼以及法律議題之關係。.....	3
圖 2，開放原始碼與單純提供原始碼、無償取得軟體之關係。.....	14
圖 3，Embedded Linux 程式授權流程。.....	21
圖 4，Android 開放原始碼計畫之系統架構與所對應之開放原始碼授權條款。	71
圖 5，MeeGo 開放原始碼計畫之系統架構與所對應之開放原始碼授權條款。	72



表目錄

	頁次
表 1，軟體授權模式之比較。.....	15
表 2，受訪談對象列表。.....	79
表 3，晶片設計公司與系統開發廠商之系統程式來源，與提供原始碼之方式。	83



第一章 緒論

嵌入式產品（Embedded Products）的開發涉及了技術面、產業面等不同的主題，而開放原始碼（Open Source）的議題亦涉及了法律、文化、經濟等範疇，嵌入式產品與開放原始碼兩者的交集更形成了相當錯綜複雜的關係。

過去與開放原始碼授權條款相關的討論，大多著眼於以工作站系統或個人電腦（Personal Computer，簡稱為 PC）作為應用平台的情形，討論在該等應用平台上，因為使用了開放原始碼所衍生的爭議與訴訟。隨著嵌入式產品功能的不斷精進，許多嵌入式產品的系統開發模式逐漸複雜化，進而仿效個人電腦的開發，使用了較為進階的軟體開發技巧與概念。也因為如此，過去曾經在工作站與個人電腦等應用平台衍生問題的開放原始碼議題，也逐漸在嵌入式產品的平台上發酵。

承上，作業系統（Operating System，簡稱為 OS）的使用，便是一個過去以個人電腦為主，現今也被應用至許多嵌入式產品的一個例子。Linux 作業系統是一套相當普及的開放原始碼作業系統，Linux 作業系統在最初發展時，雖然是以個人電腦作為執行平台，但在眾多熱心開發者的努力下，已經被廣為應用至嵌入式產品、工作站等執行平台。事實上，Linux 作業系統在嵌入式產品已占有相當領先的地位，無論是基於產品時效的考量，或以市場佔有率與支援程度為出發點，嵌入式產品開發廠商常會選用 Linux 作業系統來搭配自家的產品，此種針對嵌入式產品應用的 Linux 作業系統稱為「Embedded Linux」。

本研究選擇 Embedded Linux 作為研究主題係基考量於：Linux 作業系統在嵌入式產品中具有主流地位，且 Linux 作業系統採用的是相當具爭議性的 GPLv2 授權條款。也就是說，嵌入式產品廠商選擇使用 Embedded Linux 時，固然可以利用 Embedded Linux 來縮短產品的開發時程。但在另一方面，使用 Embedded Linux 時，也須面臨 GPLv2 授權條款具有較強的拘束力，以及規範的義務較多的兩難局面。因此本研究便欲探討在這樣的限制中，我國的嵌入式產品廠商究竟該何去何從。

附帶一提的是，細究「自由軟體（Free Software）」與「開放原始碼」的用語時，兩者的含意其實有著細微的差異，前者強調的是軟體的自由使用，後者則指提供原始碼，並使其自由流通。自由軟體係以限制性授權的方式，讓軟體得以被自由使用，即使商業公司欲從中獲利而參與自由軟體的開發時，也不至於影響被授權人的使用自由。就字義上來看，開放原始碼係與封閉原始碼相對，封閉原始碼指的是傳統商業軟體的開發模

式，也就是說，使用者在購買軟體時，只能取得包裝好、無法進行修改的軟體；而開放原始碼則提供原始碼作為檢查、修改與交互審查使用。兩相比較時可以發現，在提供原始碼的情況下，使用者使用軟體的自由仍然可能受到限制，這也是兩者的主要差別¹。

就多數情況下，自由軟體與開放原始碼二者的涵蓋範圍大致相同，因此本研究不擬區分其間的差異²，本研究中主要使用意涵較為廣泛的開放原始碼一詞，但在後續說明中，仍可能交替使用自由軟體與開放原始碼二者。

1.1 研究動機

我國的科技產業素以硬體製造見長，軟體的開發相對受到忽略。在晶圓代工的生產模式下，許多廠商專注於各類型的控制晶片設計。在開發控制晶片的同時，晶片設計公司也需要開發相對應的軟體，透過軟體來搭配控制晶片的硬體功能，方能實現功能完整的嵌入式產品。簡言之，相較於傳統的硬體生產模式，開發嵌入式產品時，廠商必須投注更多的心力在軟體功能的開發。

在開放原始碼運動萌芽時，嵌入式產品的開發平台並非開放原始碼的主要舞台，但時至今日，由於開放原始碼不需要付費即可取得，有廣大的社群資源作為後盾，又能縮短產品的開發時程，讓許多嵌入式產品的開發廠商對於開放原始碼深感興趣。然而，開放原始碼的使用雖然不像商業軟體般受到層層保護與限制，但是廠商在使用開放原始碼時，仍需受到開放原始碼授權條款的規範。

根據前述的說明可以發現，如何在符合開放原始碼授權條款的規定下，利用開放原始碼來輔助產品的開發，讓我國的嵌入式產品開發廠商可以在享受開放原始碼所帶來的便利時，避免開放原始碼授權條款可能伴隨的爭訟，對我國眾多的嵌入式產品開發廠商而言，是一個相當重要的課題。

目前在美國、德國等地，已經發生過多起與嵌入式產品、開放原始碼授權條款規範相關之糾紛，其中亦有台灣廠商涉入，因此我國的嵌入式產品廠商在使用以開放原始碼輔助產品開發時，實應特別留心開放原始碼授權條款的規範內容。為此，本研究便以嵌入式產品在開發過程中，使用開放原始碼時，可能衍生的法律問題進行探討。

¹ Andres Guadamuz Gonzalez, *Viral Contracts or Unenforceable Document? Contractual Validity of Copyleft Licenses*, Vol. 26, Issue 8, E.I.P.R. 331,339 (2004).

²自由軟體/開放原始碼的推展呈現多頭馬車的現象，其中最具代表性的便是將軟體的自由使用視為一種信仰而有著極高狂熱的自由軟體基金會（Free Software Foundation，簡稱為FSF），以及抱持較為持平論調的開放原始碼組織（Open Source Intuitive，簡稱為OSI）。前者提倡軟體的四大自由(研究、改作、重製、散布)而衍生著佐權(Copyleft)的概念，後者則以較通融、妥協的方式來看待原始碼的開放，與商業應用。

個人自大學時代便與嵌入式系統結緣，過去曾擔任嵌入式產品的排程軟體開發與系統開發職務，在進行硬體功能驗證、軟體功能、使用者介面開發等工作時，無論是系統程式的原始碼，或是輔助系統程式的開發而使用的周邊資源，如：工具鍊(toolchain)、程式庫等，均相當仰賴開放原始碼的一個重要社群，GNU，的分享，因此曾是開放原始碼的直接受益者。再者，由於修讀科技法律的關係，對於開放原始碼可能產生的法律議題感到好奇，也產生想要進一步瞭解的動力，故著手此方面的討論。

請參見第一圖，其係本研究針對嵌入式產品、開放原始碼以及法律議題所共同涵蓋的部分進行討論之示意圖。簡單來說，嵌入式產品、開放原始碼的使用，以及法律這三個看似獨立的議題間的交會處便是本研究的討論重心。此外，由於嵌入式產品的開發可能使用的開放原始碼相當多種，本研究將以 Embedded Linux 的相關授權規定作為討論的主軸。

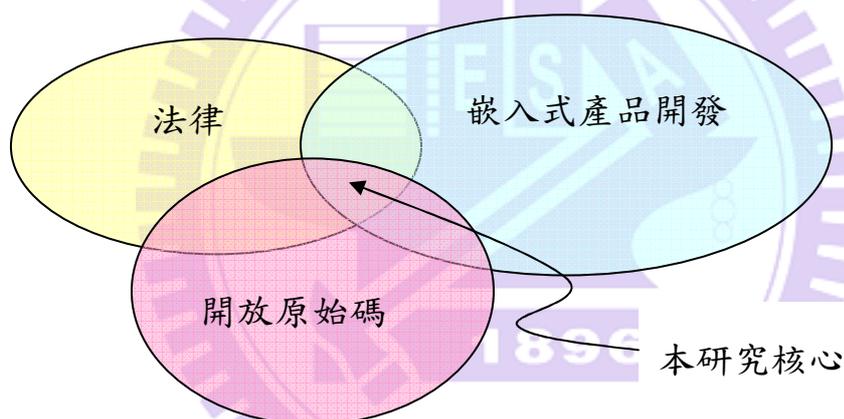


圖 1，本研究主題與嵌入式產品、開放原始碼以及法律議題之關係。
資料來源：作者整理

1.2 研究目的

一般人對於嵌入式產品一詞並不熟悉，事實上，當代的消費性電子產品如：手機、多媒體播放器、全球定位系統等，均屬於應用較為複雜而具有進階功能的嵌入式產品。在這些功能較為繁複的嵌入式產品中，為了滿足多樣化的功能需求，嵌入式產品內部的系統程式經常選擇以多工（多執行緒）的方式³來實現。由於系統程式的控制流程會較繁

³單工：作業系統一次只處理一件事情，必須等到第一件事情被完成後才會接著執行第二件事情。

多工：作業系統將要執行的事情加以切割，作業系統可以很快速的在各個功能中進行切換，由於時間間隔很短，一般使用者感覺上系統是在同一段時間內執行不同的功能。

複，爲了達到較佳的系統資源管理，與節省產品的開發時程等目的，廠商通常會利用現有的作業系統來管控系統程式。

嵌入式產品在開發時，系統程式的原始碼可能透過不同方式與來源取得，這些來源包含：廠商自行開發，或購買（授權）部份、全套商業版本的原始碼，在開放原始碼相當普及的今日，嵌入式產品的開發廠商還多了使用開放原始碼的選項。當然，就實際應用而言，廠商的原始碼可能透過不同方式取得，也就是說，同一個嵌入式產品的系統程式可能部分來自廠商自行開發、部分來自商業授權、部分爲開放原始碼。

在嵌入式產品可能使用的各種開放原始碼中，作業系統因爲與產品的特殊性較無密切關係，因此其用途較具一般性，儘管市面上使用開放原始碼授權條款的作業系統眾多，但在 Linux 作業系統的席捲下，Linux 作業系統幾已成爲嵌入式產品廠商在選用作業系統時的首要選擇，因此本研究欲對 Embedded Linux 在我國產業界的使用情況加以探討。作業系統對其上層的應用程式進行系統資源的管理，對其下層的硬體則進行驅動控制，是嵌入式產品的系統程式中，相當關鍵的一環。因此廠商所使用的作業系統若採用開放原始碼授權條款時，對於該開放原始碼授權條款內容便有必要加以瞭解，方能避免後續產生風險。

開放原始碼最初起源於個人電腦、工作站等應用平台，也曾引起相當多爭議，與開放原始碼授權條款相關的論述與判決亦所在多有，但針對嵌入式產品的討論，目前尚不多見，因此本研究將著重於開放原始碼在嵌入式產品上的應用。再者，現有關於開放原始碼授權條款的論述大多爲國外著作，並往往以開放原始碼的開發者（授權人）之立場來論述，對於身居開放原始碼授權條款中的被授權人立場的台灣廠商而言，其角色爲彼此對立的情況。我國對於開放原始碼授權條款的討論，往往以開放原始碼授權條款的法律定位、授權條款所規範的內容、與現行法律制度相關的法律議題進行分析⁴，對於實務上使用開放原始碼的情形，與開放原始碼授權條款之實際應用與否的討論仍不普及。在出發點不同的考量下，該些討論對於我國廠商並不適合一體適用，因此本研究認爲有必要針對被授權人的觀點另作討論。

本研究欲探討我國的嵌入式產品開發廠商，在發展嵌入式產品的系統程式時，利用開放原始碼輔助其研發流程之情形。因爲網路的發達，許多採用開放原始碼授權條款的軟體，往往可以輕易地透過自由、免費的方式而下載取得，一般的廠商僅知道可以在不

4陳人傑，「開放原始碼授權條款之法律分析-以通用公共授權爲中心(上)」，科技法律透析，第 14 卷第 6 期，頁 46-52（2002）。

陳人傑，「開放原始碼授權條款之法律分析-以通用公共授權爲中心(下)」，科技法律透析，第 14 卷第 7 期，頁 48-58（2002）。

需付費的情況下取得並使用該等開放原始碼，但是自由使用原始碼與免費使用軟體二者，其實有蠻大的差異，在廠商對開放原始碼授權條款的規範內容感到陌生的情況下，自然無法看出其中的端倪。

對於嵌入式產品的開發廠商而言，若能透過他人的幫助而取得現成、可用的原始碼，進而縮短自身產品的開發時程，便如同踏在巨人的肩膀上而可大幅節省開發成本。正因開放原始碼的使用相當普及，而開放原始碼授權條款又具有法律效力，對於講求開發時程的產業界而言，若只著眼於使用開放原始碼所帶來的時程上的效益，而忽略了使用開放原始碼後所伴隨的義務與授權條款的規定，將成爲一個相當值得重視的法律議題，本研究便嘗試對此進行討論。

台灣廠商素以硬體發展見長，但也因此容易忽略軟體開發的重要性，而開放原始碼的開發在某種程度上，讓台灣廠商可以降低軟體的開發成本，更讓台灣廠商相對的依賴開放原始碼。與專利訴訟相比，我國廠商對於開放原始碼的議題相對感到陌生，事實上，我國的 D-link⁵與合勤科技⁶均曾經或正在涉入與開放原始碼授權條款相關的訴訟中，這也凸顯了開放原始碼的使用已然成爲我國廠商不應漠視的議題。本研究希望能在現行的開放原始碼授權條款的機制下，對於台灣廠商使用開放原始碼的情形提出建議。

過去，多數的開放原始碼授權條款的規範往往以純軟體的應用爲出發點，而未針對嵌入式產品這種以硬體銷售爲目的的使用情形與應用加以考量，也因此造成嵌入式產品的開發廠商，在使用開放原始碼時可能發生窒礙難行的情況。綜上所述，本研究的一個目的是，探討嵌入式產品的系統程式在使用開放原始碼時，可能因爲程式的原始碼來源不同所造成的授權規定衝突，或是廠商基於利益考量而未遵守開放原始碼授權條款所衍生的法律問題，本研究另外一個目的則是，嘗試尋找能夠兼顧廠商開發權益，並維護開放原始碼基本精神的解決方法。

1.3 論文架構

開放原始碼授權條款的種類很多⁷，不同的開放原始碼也往往搭配使用不同的授權條款，由於開放原始碼的用途很多元，導致開放原始碼的使用與嵌入式產品的開發，產生

⁵ 該訴訟之非官方英文版翻譯，參見：JBB Rechtsanwälte, http://www.jbb.de/judgment_dc_frankfurt_gpl.pdf (last visited Mar. 7, 2011)。

⁶ 從 BusyBox 案談起：台灣業者侵權利用自由軟體所面對的法律風險：
<http://www.openfoundry.org/news/2277> (最後點閱時間：2011 年 3 月 7 日)。

⁷ 截至目前，由 OSI 認證通過的開放原始碼授權條款超過 60 種，參見：Licenses by Name, <http://www.opensource.org/licenses/alphabetical> (最後點閱時間：2011 年 3 月 7 日)。

許多值得深究的議題。本研究爲了將主題聚焦，以 Linux 作業系統爲範例，探討嵌入式產品的開發廠商在使用 Linux 作業系統時，因爲 GPLv2 授權條款的規範而衍生的相關問題。期待藉由此文來拋磚引玉，讓更多參與者投身相關的研究，進而讓我國產業界在使用開放原始碼時，能夠對開放原始碼授權條款有更充分的認識。本研究的內容共分八章，以下是各個章節的規劃與簡要說明。

本章爲緒論，主要說明研究動機與目的，並說明本研究的內容架構。

本研究的第二章爲研究限制，此章節說明本研究在資料蒐集與研究方法上的限制。

接下來，本研究的第三章說明了與本研究相關的技術背景，首先說明軟體的分類，藉以說明開放原始碼的特徵，其次則對嵌入式產品的系統程式架構，與作業系統在嵌入式產品的角色加以說明，最後則比較嵌入式產品與個人電腦的差別，藉此說明本研究的特色以及進行本研究的必要。

本研究的第四章爲開放原始碼授權條款的意含與法律定位，本研究對開放原始碼授權條款目前在歐美的現況進行文獻分析，其中包含：開放原始碼授權條款的背景、法律定位、主要類型及開放原始碼授權條款所造成的影響。

第五章是針對 Linux 作業系統的授權範圍，與應用 GPLv2 授權條款時的情形，並就 GPLv2 授權條款中，嵌入式產品特別容易違反的幾個條款，分別就條款的內容以及廠商違反的態樣加以闡述。

第六章主要說明 Embedded Linux 所涉及的開放原始碼授權條款的爭議與衝突，前者指的是廠商以不違反授權條款的方式規避 GPLv2 授權條款所規範的義務，後者則簡要介紹目前與開放原始碼授權條款相關的爭訟。

本研究的第七章爲係以質性研究中的深度訪談對嵌入式產品的開發者進行研究，期能透過對嵌入式產品開發過程中的前線參與者，即，開發嵌入式產品的系統工程師，瞭解我國嵌入式產品開發廠商在開發產品時，是否使用開放原始碼，以及我國廠商面對開放原始碼授權條款的態度。

本研究的第八章爲建議、結論與未來展望，首先依據嵌入式產品的特點，嘗試對 GPLv2 授權條款在嵌入式產品應用的情形提出修改建議，並在現有的 GPLv2 授權條款的架構下，提出幾點建議，作爲嵌入式產品開發廠商在使用開放原始碼時的參考，最後則對未來研究的方向與議題提出些許建議。

第二章 研究限制

台灣的科技業廠商以硬體開發見長，對於軟體開發所投注的心力相對較少，也因此，在開放原始碼的議題中，我國廠商扮演的往往是開放原始碼的被授權人而非授權者。參與開放原始碼的開發者往往散居世界各地，其身分也難以識別。基於時地的考量，本研究選擇以開放原始碼的被授權人為出發點，至於授權者之看法與論點，則以參考文獻為主。因此，本研究受限於我國廠商的先天特性與有限的人力資源，未能以兼容並蓄的方式探討對立兩端的看法，此為本研究最大的限制。

2.1. 研究範圍的限制

誠如前述，由於嵌入式產品與個人電腦有諸多本質上的歧異，本研究認為有必要特別針對這類型的議題加以探討，而無法直接使用參考文獻的資訊。然而，即便將焦點放在以嵌入式產品的平台為主的開放原始碼，衍生的議題仍然很多，因此本研究將主題進一步限縮在使用 Linux 作業系統的嵌入式產品，並以 Linux 作業系統所使用的 GPLv2 授權條款作為討論的核心。此外，儘管過去對於軟體的相關法律議題有蠻多的爭議，但本研究將焦點著重在以開放原始碼授權條款進行授權的程式，因此關於程式在著作權、專利的法律定位，以及與個人電腦、工作站平台相關的開放原始碼之訴訟與爭議便不特別論述。

2.1.1. 選擇 Embedded Linux 作為討論對象

嵌入式產品的系統開發包含許多環節，而如何選擇開放原始碼，並應用於系統中，也涉及許多層面的考量。在眾多的系統程式中，作業系統位居資源管理的要角而提供承上啓下的功能，對於上層的應用軟體與下層的硬體驅動功能都相當重要，因此以使用開放原始碼的作業系統作為討論主體，可以說明開放原始碼對於系統程式的影響程度。再者，在眾多的開放原始碼作業系統中，Linux 作業系統具有相當重要的地位，為了將研究主題聚焦，本研究僅以使用 Linux 作業系統的嵌入式產品，即，Embedded Linux，的情形加以討論。

2.1.2. 開放原始碼授權條款的選擇

由於開放原始碼授權條款的種類繁多，本研究無法兼顧所有與嵌入式產品開發相關的開放原始碼授權條款，而是將相關的討論環繞在 Linux 作業系統所選用的 GPLv2 授權條款，此為本研究研究的另外一層限制。

2.1.3. 受訪對象的選擇

本研究的訪談研究以利用開放原始碼的被授權人為主，即嵌入式產品開發者為討論對象，而未針對開放原始碼的授權人（參與開放原始碼計畫的社群）進行訪談。最主要的原因是，開放原始碼的相關計畫經常透過在世界各地的程式設計者一同開發，對於開發者的身份不易掌握，也不容易找到參與開放原始碼計畫的程式設計者進行訪談。

即便台灣也有參與開放原始碼開發的社群，但是當前參與國內開放原始碼計畫的程式設計者，大多著重於網路軟體與個人電腦應用軟體的開發，該些軟體與本研究所著眼的嵌入式產品所搭配的系統程式不同，導致該些程式開發者並不適合擔任本研究的受訪者。易言之，本研究在進行受訪對象的選擇時，係以開放原始碼的需求端為著眼點，關於供應端的部份，受限於外在環境的考量而未能予以探討，以至於本研究的研究未能同時呈現對立兩端的看法。

2.2. 資料蒐集的限制

過去與開放原始碼相關的爭議係以個人電腦及/或工作站平台為主，目前與 Embedded Linux 直接相關的爭議案件相當有限，然而，嵌入式產品在本質上有許多與傳統的個人電腦平台、工作站平台不同之處，使得過去爭議的內容不適合直接被套用至嵌入式產品的情況。換句話說，直接與本研究主題相關的資料不多，因此本研究必須根據有限的資料來分析，或是透過類推的方式，將過去在個人電腦及/或工作站平台的論點，套用至嵌入式產品的情況。

我國國內目前尚未發生與 Embedded Linux 相關的案件，但我國的嵌入式產品開發廠商，曾經因為所販售的產品，並未依照 GPLv2 授權條款的規定提供原始碼，並因此而涉入相關的爭訟。無論是晶片設計公司或系統開發廠商，台灣多數的嵌入式產品開發廠商大多屬於開放原始碼的被授權人，由於嵌入式產品本身具有個別化應用的特性，各家廠商在發展嵌入式產品時，其作法也不一定相同，這也讓嵌入式產品使用開放原始碼情形

之相關資料較不容易取得。

在自由軟體鑄造場⁸等組織的努力下，台灣也有一些開放原始碼計畫在進行中，進一步瞭解後，可以發現這些本地的開放原始碼計畫係以網路應用為主，少有針對嵌入式產品的應用，遑論廠商是否會因為使用了開些原始碼，而產生我國國內的社群與廠商間發生爭訟的可能，也因此，較不容易發生我國廠商使用了我國的社群所開發的開放原始碼，而於我國的法院衍生爭訟的情形。

在網路普及的今日，開放原始碼的專案往往透過世界各地的社群來完成，因此開放原始碼計畫的著作權人也散居世界各地而不易追尋，儘管如此，開放原始碼的相關訴訟並不因而無法進行。目前與開放原始碼相關的訴訟係以其中較為激進者所提出，例如德國的 Harald Welte 與美國的軟體自由法律中心（Software Freedom Law Center，簡稱為 SFLC）。換言之，與開放原始碼授權條款相關的訴訟容易成為跨國訴訟，因此無法特別針對國內發生開放原始碼訴訟的情形討論。

2.3. 研究方法的限制

2.3.1. 文獻分析法的限制

本研究的研究方法使用文獻考察與訪談，針對文獻分析的部份，儘管目前在德國與美國有幾個相關與 Linux 作業系統或嵌入式產品相關的判決，但其中與 Embedded Linux 直接相關的案件尚不多見，另一方面，受限於本研究作者的語言能力，僅能以英文與中文的相關文獻進行資料蒐集與討論，因此關於德國法院的判決，作者無法直接閱讀第一手資料，僅能參考英文的學術文章與新聞。

2.3.2. 半結構式訪談的限制

除了無法對開放原始碼的開發者進行訪談的限制外，由於本研究涉及公司內部系統開發作法等細節，可能涉及公司與員工所簽訂的保密協定等，因此在決定訪談對象時無法對各嵌入式產品的開發者先進行抽樣後才訪談，而是由研究者在考量與研究主題的關聯性與相關年資等而選定，因此可能會有代表性不足的問題。

⁸ 中央研究院資訊科學研究所所設置的自由軟體鑄造場(Open Source Software Foundry, OSSF)除了提供自由軟體相關的法律諮詢服務外，更主動在國內推廣各種自由軟體計劃的開發與應用。



第三章 技術背景

3.1 嵌入式產品

簡單說來，嵌入式產品是透過一個小型微控器控制相關電路的系統，而手機、數位相機、汽車內的微電腦、各類型家電、導航系統…等均屬於嵌入式產品⁹，嵌入式產品不需要像個人電腦一樣使用功能強大的中央處理器（Central Processing Unit，簡稱為 CPU），程式操作流程也較為簡易，因此一般的產品使用者（消費者）不會意識到這些產品需要使用控制晶片來提供各種操作功能。

3.1.1. 嵌入式產品的硬體架構與產業分工

根據前述的說明可以發現，許多消費性電子產品均屬於嵌入式產品的範疇，目前與消費性電子產品的生產過程相關的廠商，可以根據產品的不同而區分為晶片設計公司、系統開發廠商與品牌廠商三類。簡單來說，消費性電子產品常見的開發模式是，晶片設計公司為了證明自己所販售的控制晶片功能正常，常需要將控制晶片所需的軟體功能一併開發完成，讓系統開發廠商可以直接使用，也可以節省系統開發廠商的開發時程，而系統開發廠商的客戶為品牌廠商，無論是軟體或硬體，品牌廠商鮮少對嵌入式產品的內部加以改動，而是要求系統開發廠商配合，由系統開發廠商完成品牌廠商所要求之功能選項或操作介面。

也就是說，目前嵌入式產品的系統程式，往往由對晶片功能較為了解的晶片設計公司來提供，而系統開發廠商相當於晶片設計公司的後手，兩者均可能利用開放原始碼來輔助系統程式的開發，而品牌廠商則因較少參與系統程式的開發，故不在本研究討論之列。

承上，嵌入式產品的系統開發過程，可以依據生產的階段而概括區分為：晶片設計與系統開發兩種階段。在晶片設計階段，晶片設計公司設計並販售嵌入式系統中，主要用來執行核心操作的控制晶片；在系統開發階段，系統開發廠商向晶片設計公司購入控

⁹ Rick Lehrbaum, *The Linux Devices Showcase*, <http://www.linuxfordevices.com/c/a/Linux-For-Devices-Articles/The-Linux-Devices-Showcase/>, (last visited Jan. 17, 2010).

制晶片，之後再搭配不同的軟體功能與硬體元件來販售。需注意的是，無論嵌入式產品的種類為數位電視、手機或其他產品，這種依據生產階段的分類方式並不會因為其功能或產品的不同而異，因此以下將以這兩種類型的硬體開發廠商作為討論的對象。

3.1.2. 嵌入式產品的軟體架構與開發環境

在進行嵌入式產品的系統程式開發時，為了管理與維護的目的會將系統程式內的原始碼進行劃分，劃分的方式主要有分層與模組的分別，以蓋房子比喻系統程式的開發時，分層相當於不同樓層的區別，而模組則代表在各個樓層中的各個房間規劃。

在功能較為強大的嵌入式產品中，至少可將系統程式的軟體架構區分為四個層級，分別是：與硬體習習相關的驅動程式(driver)、用來控制整體系統程式之操作流程的作業系統層(OS)、提供消費者操作時的功能之應用層(Application)，以及呈現給消費者進行操作之使用者介面層(User Interface, UI)。

軟體的開發過程需要從原始碼轉換成目的碼，而嵌入式產品的相關系統資源有限，無法提供完整的除錯(debug)與編譯(compile)環境，因此，嵌入式產品在進行系統程式開發時必須透過其他應用平台(如：個人電腦)的輔助才能進行原始碼的開發，並進而產生執行的目的碼。目前最普遍的嵌入式產品的開發環境，是由 GNU 所開發出來的工具鍊(GNU tool chain)¹⁰，使用 GNU 工具鍊作為開發環境的好處是，微處理器開發廠商可以降低提供開發環境所需的研發人力，晶片設計公司與系統開發廠商，也得以用較低廉的成本取得開發環境。

經由上述說明可以發現，嵌入式產品的開發過程中，除了系統程式可能會使用以 GPLv2 授權條款所授權的 Linux 外，系統程式開發所需搭配使用的開發環境亦可能會使用由 GNU 所提供的工具鍊，也就是 GNU 的存在對於嵌入式產品廠商而言確實相當有助益。

若以做菜比喻系統程式的開發時，開放原始碼可被類比為公開、免費的食譜，而 GNU 工具鍊則相當於 GNU 所提供的公用廚房。開放原始碼授權條款要求被授權者在該免費食譜的協助下，若作出新的菜色並加以散布時，被授權人需要提供他完成該菜色之細節，讓其他的被授權人有更多的食譜可以參考；而 GNU 工具鍊則讓程式開發者在烹調前，不必另外花力氣籌備一個廚房，而能夠使用現成的廚房設備來烹調，但是程式開

¹⁰處理器核心大廠如 ARM、MIPS 在提供開發環境時，都會以 GNU 工具鍊為基礎提供客戶程式開發的環境。以 MIPS 為例，該公司所提供的開發環境均以 GNU 工具鍊為基礎，而 ARM 則同時提供一種由該公司內部發展的開發環境，以及以 GNU 工具鍊為基礎的開發環境，採用雙重管道的方式提供給客戶，其客戶便能視自身的需要而選用。

發者利用這個現成廚房所烹調的菜色並不以開放原始碼為限，也就是程式開發者可以利用 GNU 工具鍊來開發各種類型之程式，無論該程式之授權方式為商業授權或開放原始碼授權。

易言之，對於嵌入式產品開發廠商而言，GNU 不但提供了食譜，還提供了相關的設備讓使用者使用，而嵌入式產品開發廠商得以利用免費提供的環境，以及免費食譜所提供的資源，發揮其巧思來進行系統程式的開發。

3.1.3. 嵌入式產品的作業系統

作業系統可以被簡單形容是一個協調系統資源的平台，特別是對於系統程式中，所需要執行的功能較多而需透過多工處理的情形。也就是說，在單工處理無法滿足系統需求時，便常常透過多工處理的作業系統來進行資源分配。

對於一般使用者而言，作業系統並不影響使用者的操作介面，而只是一個用來提供上層軟體運作的資源管理程式而已。因此，在上層的應用軟體相同，且作業系統均可以支援上層的應用軟體時，一般使用者並不會因為作業系統的不同而感到差異。在個人電腦平台上，微軟的視窗作業系統具有獨占地位，這是因為許多應用程式都僅支援微軟的視窗作業系統，而不支援其他作業系統的關係，久而久之，多數的使用者便相當倚賴微軟的視窗作業系統。

隨著嵌入式產品的複雜化，當前許多嵌入式產品也導入了作業系統的作法，然而作業系統的開發相當費時，因此多數的廠商不會自行開發專用的作業系統，而是採用市面上可得之作業系統以節省產品的開發時程。與個人電腦上以微軟的視窗作業系統為大宗的情形不同，嵌入式產品所使用的作業系統種類相當多元，且根據授權費用的收取與否而可分為需要收取授權費用與不需要收取費用兩種類型。需要授權費用的作業系統有：VxWorks 作業系統、Palm OS 作業系統、Windows CE 作業系統、Nucleus 作業系統等；而不需要收取授權費用的作業系統則有 Linux 作業系統、eCos 作業系統、uClinux 作業系統等。

在前述兩種類型之嵌入式產品的作業系統中，前者屬於商業化銷售的產品，採用此類型的作業系統時，系統程式的開發者不用擔心作業系統內部的管理，對於公司在開發新一代產品或不同系列產品時，具有容易取得授權公司的支援，以及容易移植應用軟體等優勢，若廠商對作業系統的程式有疑問時，也有明確的被諮詢對象；而後者的優點則是廠商在使用時，不需要支付軟體授權費用，該些作業系統係以公開原始碼的方式提供，讓廠商的取得成本大幅降低。

由於開放原始碼容易被散布，相關的口碑也較容易流傳，在成本較低且資訊透明化的考量下，嵌入式產品使用開放原始碼的情形也越來越普遍。但是廠商在使用開放原始碼的作業系統時，還需要進一步考慮作業系統是否提供足夠的支援服務，廠商使用開放原始碼的作業系統時，一旦遭遇問題，並沒有明確的諮詢對象，也不一定能仰賴社群來解決，而必須由廠商自行克服，因此廠商的技術能力也是影響作業系統選擇的關鍵因素。

無論嵌入式產品的使用的是何種作業系統，與個人電腦上的作業系統相當不同的是，個人電腦上的作業系統選擇很少，而且一般使用者可以清楚知道自己所使用的作業系統種類，而應用於嵌入式產品的作業系統種類卻多如牛毛，就一般使用者而言，在使用作業系統時，作業系統的存在與否，以及使用的作業系統種類等問題，並不會影響一般使用者操作的流程。

3.2 軟體的分類

電腦能夠執行各類型操作的原因就是透過軟體的各種應用，除了一般常見商用軟體如：微軟的視窗系統、辦公室系列等需付費才能使用的軟體外，近年來如防毒軟體、即時通訊軟體、遊戲軟體等，也經常以免付費軟體的形式出現。需注意的是，不需要付費的軟體僅代表軟體的取得無須支付費用，與原始碼是否被提供不同，因此免付費軟體並非本研究所討論的開放原始碼。

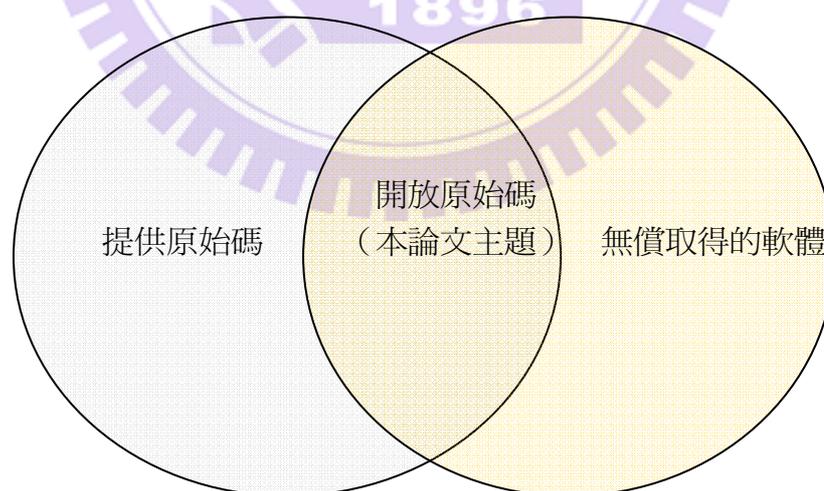


圖 2，開放原始碼與單純提供原始碼、無償取得軟體之關係。

依照被授權人取得軟體時是否需要支付授權費用來分類時，軟體的授權契約可以分

為：有償契約與無償契約¹¹，直覺上，採用無償契約授權的軟體似乎與付費軟體相對應，即，免費即可取得並使用的軟體。實際上，採用無償契約授權的軟體還可再依照原始碼提供與否，而被進一步再區分為兩種類型，第一種類型與商業軟體類似，即不提供原始碼給被授權人，只是在提供軟體時，軟體開發者不對被授權人索取授權費用；第二種類型則是在提供軟體（目的碼）的同時，一併提供程式的原始碼給被授權人，讓被授權人可以對原始碼進行修改。

另一種分類方式，則是根據原始碼的提供與否，將軟體區分為專屬軟體（proprietary software）以及非專屬軟體（non-proprietary software），前者包含一般需要付費使用的商業軟體，以及雖然不需要付費卻無法取得原始碼的免付費軟體，與功能受到限制的共享軟體；後者則指將程式的原始碼，無償提供給被授權人使用的開放原始碼。簡單來說，後者希望提供給被授權人的程式使用自由，往往是前者利用授權條款來規範，並限制被授權人不得擁有的權利。

下表是對各種軟體授權模式的簡單分類：

表 1，軟體授權模式之比較。

	有償契約	無償契約	
不提供原始碼	常見的商業軟體 ¹² (專屬軟體)	以雙重模式銷售 的免費軟體	透過廣告營收獲利的 免費軟體
		共享軟體 試用軟體 ¹³	透過廣告收益的軟體 ¹⁴
提供原始碼	授權開發使用 ¹⁵	開放原始碼（自由軟體）	

資料來源：作者整理

若以蓋房子比喻軟體的授權時，商業軟體相當於一棟由商業軟體廠商蓋好後，被授權人必須付費購買後方能居住、使用的房子；而免費軟體則相當於一棟程式開發者蓋好

¹¹ 李憲隆，「開放原始碼法律問題之研究-智慧財產權制度之檢討」，中大產經所碩士論文，頁 23-24(2003)。

¹² 一般的商業軟體公司所販售的產品，如：office、windows。

¹³ 亦由商業軟體公司提供，可能會依據用途不同而提供不同版本的軟體，或是提供有限時間或使用次數版本的軟體，如：Avira AntiVira 防毒軟體。

¹⁴ 透過使用者在操作軟體時所衍生的其他效益而獲利的商業軟體，如：MSN。

¹⁵ 廠商與廠商之間透過軟體授權條款的簽訂，需支付軟體授權金，如：Windows ME。

後，免費提供給被授權人居住、使用的房子，或者像是免收門票的公園，提供空間給參觀者使用，但是會受到開放時間、功能的限制等。也就是說，商業軟體與免費軟體的差別為：被授權人在使用軟體時，是否需要付費。兩者的共通處則是，被授權人未能參與建物的搭建，而是全然的接受搭好的建物，因此對於建物的外觀設計、內部陳設、空間規劃等，被授權人均無法分享其想法。

另一方面，提供原始碼給被授權人的作法，可以被視為提供房子結構之設計圖與施工藍圖給被授權人，在開放原始碼的情況下，被授權人(住戶)可以視個人需要，依據開發者所提供的資訊(原始碼)自己按圖索驥，對房子的結構進行修改；而前述的商業軟體與免費軟體則代表建商(開發者)在房子蓋好後，分別以有償與無償的方式僅將房子交給住戶(被授權人)，卻同樣未提供設計圖給住戶(被授權人)。

對被授權人而言，是否提供(開放)原始碼，相當於建商是否願意提供房屋之藍圖給住戶(被授權人)，這也代表被授權人是否能對房屋的整修或擴建發表個人意見，或是親身參與房子的搭建。若建商並未提供藍圖，被授權人一旦需要改進程式的功能時，自然不知道從何著手，類比至程式的開發時，被授權人便無法對組成該軟體的原始碼進行修改。或許在不提供藍圖的作法時，對於一般的被授權人，也就是一般的程式使用者已經可以滿足其需求，但是對於想要讓軟體具有更多功能，或對於軟體之使用必須受限於軟體提供廠商而感到厭煩的玩家而言，這樣的作法則無法滿足他們。

綜上所述，不難發現，商業軟體授權的目的是排他，因此所要保護的是不讓他人得以製作、生產相同的成品，而開放原始碼授權條款則強調分享，因此希望公開製作過程，讓大家都能有機會做出相同的成品，甚至提供、分享改進的方式，而讓軟體的功能更為強大。易言之，商業軟體授權條款與開放原始碼授權條款所欲保護的標的截然不同，前者保護的標的是可執行的軟體，也就是末端的成品，商業軟體的開發則是為了販售，在商業軟體的開發過程中，原始碼均受到相當嚴密的保護，因此，能夠接觸商業軟體之原始碼的人，僅限於商業軟體公司內部參與開發的員工。而後者強調的是「程式原始碼」可以被自由使用的授權，因此，根據這兩種模式而開發的軟體也因為訴求本質的不同而產生在後續使用時的差異。

3.3 嵌入式產品與個人電腦的差別

儘管嵌入式產品可被類比為功能較為精簡的個人電腦，但是嵌入式產品與個人電腦的用途與設計仍有相當多不同的地方，而這些本質上的差異，也連帶使過去對於開放原始碼的相關討論，並不適合直接套用至嵌入式產品，為了能清楚比較其差異以利後續的

討論，在此簡要以使用場合、數量、使用者對於軟體與硬體的感受程度、廠商的獲利來源、產品的相容性與開發模式等面向，區分個人電腦系統與嵌入式產品的主要差異。

3.3.1. 使用場合與數量

當使用者在操作電腦時，會相當清楚自己正在使用電腦，但是嵌入式產品因為隱身於日常生活中，而讓使用者未意識到嵌入式產品的存在。在一般的家庭中，個人電腦的數量大多為在個位數，相較之下，一般家庭中少則十來個，多則存在數十個甚至上百個嵌入式產品。

個人電腦所提供的功能相當明確，使用者透過電腦進行文書操作、上網、觀看影音節目、玩遊戲等，使用者都相當確知自己所要操作的功能。反觀嵌入式產品的應用則五花八門，例如家中成員的手機，家中的電視、冰箱、洗衣機、冷氣機等家電用品，甚至汽車、電子錶等，都屬於嵌入式產品的應用範圍。因此就使用場合來說，個人電腦的使用場合較為特定而有形，而嵌入式產品的使用場合則較為多元而無形。

3.3.2. 軟體與硬體的感受程度

一般的使用者在使用個人電腦時，其實是對在個人電腦執行的各種應用程式(軟體)進行操作，由於個人電腦的硬體規格為統一化的標準，因此一般使用者對於硬體的差異除了執行速度的快慢外，並不會有深刻的感覺，而是對軟體的畫面與操作流程有較為深刻的印象，而該些功能係由應用軟體所提供，與硬體的相關性較低。

另一方面，嵌入式產品所提供的功能通常為專屬功能，提供給一般使用者的操作介面也相當簡單明瞭，例如數位相機與電子錶的功能便相當明確且不同，有些嵌入式產品甚至不提供操作介面，使用者並不會意識到這些功能同樣需要透過軟體來控制。

換言之，一般的使用者在操作個人電腦時，對於軟體的感受較為深刻，而使用者在操作嵌入式產品時，對於硬體本身提供的功能具有較深刻的感受。

3.3.3. 廠商的獲利來源

根據一般使用者對於軟體與硬體的感受程度不同，可以發現個人電腦的軟體廠商透過軟體的授權費用而獲利，而嵌入式產品的開發廠商，無論是晶片設計公司或是系統開

發廠商，雖然會提供軟體的功能，但提供軟體的目的並非以軟體授權來獲利，而是著眼於硬體的銷售，即肉眼可見的實體產品之販售。儘管硬體的功能需要透過軟體的控制才能正常運作，但對於硬體開發廠商而言，實際售出的產品指的是有形的硬體，即晶片設計公司所販售的晶片，以及系統開發廠商所販售的嵌入式系統產品。

對嵌入式產品的開發廠商來說，軟體非但不是銷售的主體，甚至必須免費提供維護與更新，也就是說，軟體的開發代表成本的提升，卻不一定能為廠商帶來相對應的獲利，這也讓嵌入式產品的開發廠商對於軟體功能的開發感到卻步，若能夠利用現成的開放原始碼為基礎，廠商自然會以成本較低的方案，作為嵌入式產品開發時優先考慮的對象。

另一方面，就個人電腦的生產而言，軟體開發廠商與硬體開發廠商的分野相當明確，前者如微軟，後者如華碩、宏碁等公司，這兩種廠商的專業分工相當的明顯。因此以商業軟體作為獲利來源的商業軟體廠商便著重在軟體功能的開發，即，開發軟體的目的便是為了販售軟體的授權，而不是為了販售硬體而開發軟體。而個人電腦的硬體開發廠商亦以硬體的效能與其生產為主要核心，並不會涉入軟體的開發。

換言之，嵌入式產品的開發廠商僅透過硬體獲利，卻必須承擔軟體開發的成本，而個人電腦的軟體廠商與硬體廠商則各有其獲利來源，此為這兩種類型的產品的另一個差異處。

3.3.4. 產品的相容性

若將個人電腦系統的開發比喻為鐵路系統，其軌道寬度與格式必須使用一定的規格，讓各家廠商製作出來的火車車廂都能在上面行使的話，各家廠商所製造的火車車廂與輪軸間距等亦需要遵守一定的規格，否則其火車車廂在生產後也無法行駛。而嵌入式產品的系統開發則類似汽車廠商對於車輛的開發，只要車輛製造出來可以行動自如，其樣式與規格並不需要特別限制。

由於個人電腦的硬體系統採用的開放性、一致性的標準，只要電腦硬體的生產依照一定的規格，便可以搭配電腦軟體來使用。相較於單純生產硬體的個人電腦製造商，嵌入式產品的開發廠商需要整合軟體開發的技術，而嵌入式產品本身亦較為獨立而無法交換使用應用程式，即便有，也只是網路應用或是最上層的應用軟體，而非架構在硬體上的底層程式與作業系統等，因此搭配嵌入式產品的軟體必須針對各家的硬體量身打造，不但硬體的選用沒有一套共同定義的規格可供參照，軟體的規劃更是五花八門。

3.3.5. 系統軟體開發模式

由於個人電腦的硬體具相容性，因此相同的目的碼可在不同的個人電腦上使用、執行，而毋需進行修改。亦即，個人電腦系統的商業軟體開發廠商只要發展一套軟體，便可以在所有的個人電腦使用，因此若使用開放原始碼時，軟體廠商並不需要考慮與硬體的相容性，而是著重在開放原始碼的功能改進，更不需要考慮硬體的差異，而大幅修改取得的開放原始碼。另一方面，對嵌入式產品的開發者來說，採用開放原始碼的主要目的並不是為了精進開放原始碼所提供的功能，而是為了將開放原始碼既有的功能，套用至自家的硬體，讓自家的產品也能提供開放原始碼所達成的功效。

易言之，以個人電腦作為執行平台時，開放原始碼的可被利用性較高，因此利用開放原始碼的軟體廠商獲得助益的程度也較高；但是對嵌入式產品的開發廠商而言，系統程式的涵蓋範圍很廣，除了使用作業系統外，系統程式還包含了使用者介面、應用程式、驅動程式等，因此即便廠商直接使用 Linux 作業系統的核心程式，而完全未對 Linux 作業系統的原始碼加以修改，但是對整體的系統程式而言，開放原始碼所占的比率仍然相對較低。

3.4 開放原始碼與嵌入式產品

基於開放原始碼的互益本質，使用以開放原始碼授權條款進行授權的軟體時，被授權人並不會如商業授權軟體一樣受到諸多限制，這也讓嵌入式產品的開發廠商，得以根據產品需求的不同，而彈性的對開放原始碼進行客製化(customized)，也就是針對自家產品的需要而對開放原始碼加以改作，以增加軟體與硬體之間的契合度與穩定性。此外，由於開放原始碼的使用並不需支付授權金，讓系統軟體的開發成本相對降低，這也讓嵌入式產品的開發廠商，無論是晶片設計公司或是系統開發廠商，不由得對開放原始碼所提供的現成資源感到興趣。

經由上述的說明可以發現，就最基礎的概念而言，嵌入式產品與個人電腦雖然同樣利用軟體來控制硬體的功能，但是二者的應用範疇、開發時的考量，硬體相關性與架構設計上，仍有相當大之本質差異。當環境係以一個較為廣泛而分散的模式進行開發時，個人與小團體較能表達個人意見、交換見解，並建立他們自己的資訊環境¹⁶，由於個人

¹⁶ Benkler, Yochai, From Consumers to Users: Shifting the Deeper Structures of Regulation toward Sustainable Commons and User Access, 52, Fed. Comm. L. J. 561,583 (2000).

電腦的平台採用一致的硬體架構，一般消費者對於電腦平台所執行的程式相對容易，而能以較為廣泛而分散的模式交流其對開放原始碼的建議。但就嵌入式產品之系統程式開發而言，則因為各家廠商的作法不一，對於開放原始碼的規劃與應用也不同，一般消費者進行資訊交流所能取得之資源較為零碎，因此要結合消費者的力量，對嵌入式產品的系統程式進行修改，進而提高系統程式的品質相對較不容易，在使用者參與改良開放原始碼卻無法帶給廠商相對效益的情況下，開放原始碼的品質不一定能被提升，將進而喪失讓廠商提供原始碼的誘因。而這也是本研究認為，既有的開放原始碼授權條款，需要針對嵌入式產品的應用特別討論的原因。

易言之，本研究的考量為，嵌入式產品與個人電腦的開發模式既然有如此大的差異，若使用相同的授權條款來規範是否恰當？在個人電腦上執行的 Linux 作業系統，以及在嵌入式產品所使用的 Embedded Linux 雖然師出同門，但對被授權人而言，所需要付出的個人努力程度，以及可以因此獲得的利益也不同，使用同樣的 GPLv2 授權條款來規範被授權人是否合理？

附帶說明的是，本研究探討的廠商，是以在嵌入式產品開發過程中，使用開放原始碼作為系統程式之一的嵌入式產品開發廠商。即，廠商所銷售的嵌入式產品中，包含了以開放原始碼授權條款進行授權之軟體。至於在公司內部使用以 Linux 為作業系統的工作站，或是公司發展 Linux 套件並提供相關的技術支援服務等情形，則非本研究的討論對象。

3.5 嵌入式產品中的 Linux 作業系統-Embedded Linux

Linux 的原始作者 Linus Torvalds 最初發展 Linux 作業系統時，將原始碼提供在網路上，並以 GPLv2 授權條款進行授權，讓不同的參與者加入開發，使得 Linux 作業系統的功能與應用也越來越廣泛、越來越多元。時至今日，除了最初的個人電腦外，Linux 作業系統可應用的平台已經跨足至工作站、嵌入式產品等平台。

儘管嵌入式產品可使用的作業系統相當多元，但是 Linux 作業系統具有功能完整、取得成本低等優點，讓 Linux 作業系統的市佔率居高不下。舉例來說，針對複雜的網路應用，Linux 作業系統提供了相當完善的方案，讓許多需要支援網路功能的嵌入式產品開發廠商趨之若鶩。

在瞭解嵌入式產品的用途、開放原始碼授權條款的目的，以及 Linux 作業系統在嵌入式產品所扮演的角色後，接著利用下圖說明，嵌入式產品開發廠商在使用 Embedded

Linux 時，會因為 Linux 作業系統所採用的 GPLv2 授權條款，而遭遇到過去使用商用授權之原始碼，或自行開發系統程式的原始碼時，不曾面臨過的法律議題。

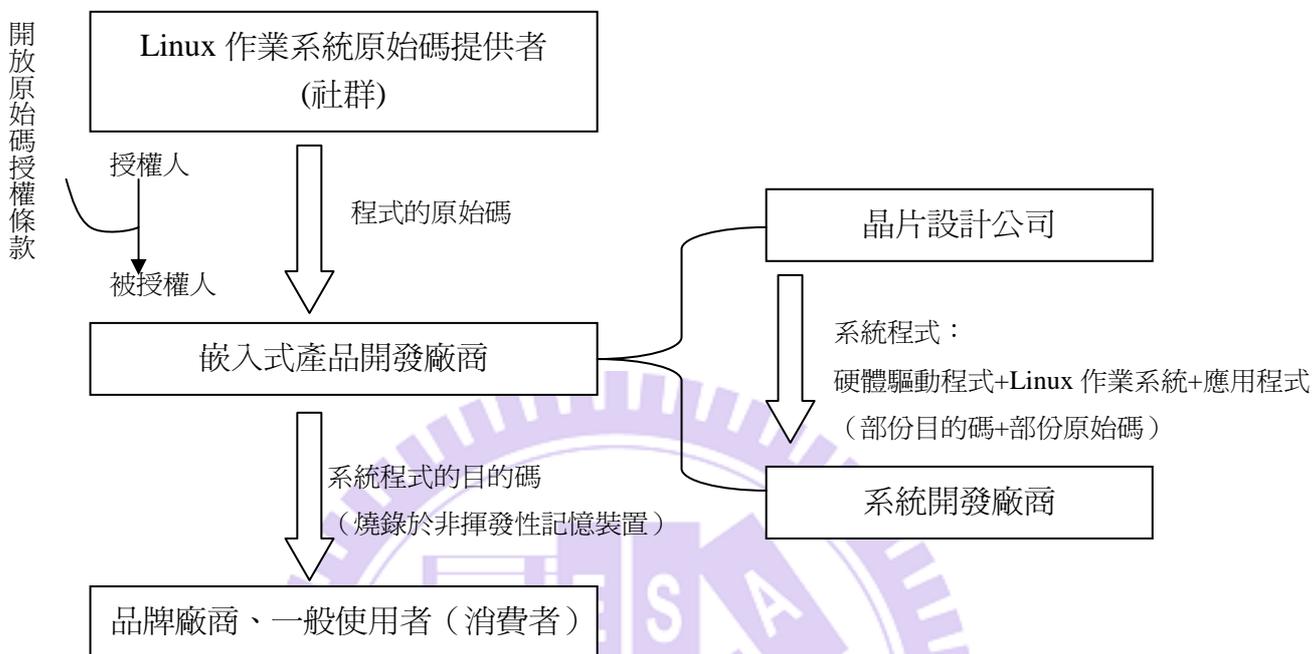


圖 3，Embedded Linux 程式授權流程。
資料來源：作者整理

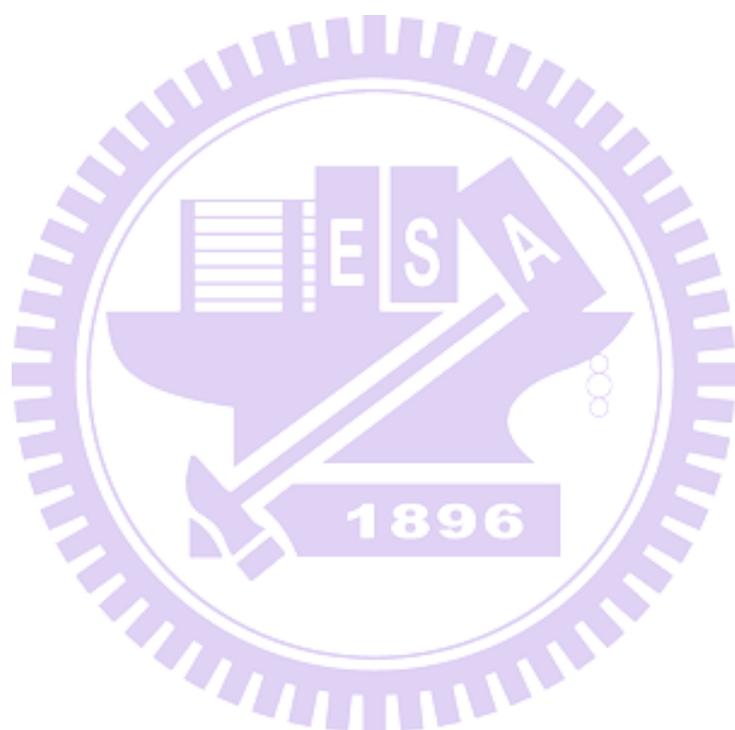
根據 Embedded Linux 程式授權流程示意圖可以看出，Embedded Linux 的程式在散布過程中，可以被概略區分為兩個階段，第一個階段指的是嵌入式產品廠商取得 Linux 作業系統，第二個階段指的則是，嵌入式產品開發廠商販售嵌入式產品給品牌廠商、一般使用者（消費者）。

Linux 作業系統的原始碼係由社群開發而產生，透過 GPLv2 授權條款而提供原始碼給被授權人，對於 Embedded Linux 而言，被授權人即為嵌入式產品的開發廠商。嵌入式產品開發廠商中的晶片設計公司，在取得 Linux 作業系統的原始碼後，將 Linux 作業系統、硬體驅動程式與應用程式結合而形成系統程式的軟體堆疊，並以部份目的碼、部份原始碼的組合方式，提供給系統開發廠商。而嵌入式產品在銷售時，系統程式的目的碼將被燒錄於內部的非揮發性記憶裝置中，再被販售給品牌廠商與一般使用者（消費者）。

圖中的第一階段，也就是嵌入式產品開發廠商取得 Linux 作業系統的原始碼時，相當於第一階段中的被授權人，而 Linux 作業系統的眾多開發者則為授權人。依照 GPLv2 授權條款的規定，Linux 作業系統的程式在複製、散布時，仍須依循 GPLv2 授權條款的授權條件。當嵌入式產品開發廠商在販售嵌入式產品時，雖然同樣符合 GPLv2 授權條款

所定義之複製、散布行爲，但目前對於圖中的第二階段，也就是 Linux 作業系統被用來搭配嵌入式產品的其他系統程式，並販售給消費者時，嵌入式產品的開發廠商是否遵守 GPLv2 授權條款之規定，以及 Embedded Linux 的散布所衍生的法律問題，更是本研究所欲探討的。

導入 Linux 作業系統，不但讓嵌入式產品的開發得以加速，亦使 Linux 作業系統的市占率提昇，二者形成了魚幫水、水幫魚的互益關係。由於 Linux 作業系統支援的功能越來越多，讓 Linux 作業系統在市場上的地位也越來越重要，造成嵌入式產品的開發廠商，無論基於技術上可得資源的考量，或是著眼於市場的接受度，都傾向使用 Linux 作業系統，因此本研究將以 Embedded Linux 作為討論的核心。



第四章 開放原始碼授權條款

目前國內關於開放原始碼授權條款的討論，大多討論開放原始碼在一般性平台之使用問題，例如：利用以開放原始碼授權條款作為授權的軟體進行教學、降低軟體取得成本而使教學普及化等討論，或是探究公司內部之個人電腦、伺服器如何利用開放原始碼來降低開銷（例如：利用 Open Office 而不使用微軟的 Office 系列軟體來進行文書處理、使用 Linux 作業系統而不使用 Windows 作業系統等），而與本研究特別針對嵌入式產品的本質進行討論不同。因此本研究中關於開放原始碼授權條款在嵌入式產品應用的立論基礎，係以國外的著作、論文、判決與相關社群網站的文章為出發點。

4.1 背景

4.1.1 開放原始碼授權條款的起源

Richard Stallman¹⁷在 1980 年代發起自由軟體運動（Free Software Movement），並成立了自由軟體基金會（Free Software Foundation）來推動 GNU 計畫，Stallman 認為，採用商用授權方式來販售軟體的作法，將限制軟體的發展，並認為應該將程式，無論是程式的原始碼(source code)或是目的碼(object code)提供給取得程式的後手，並主張程式應該保持「研究、改作、重製、散布」的四大自由，以達到資訊交流與共享的目的。Stallman 與 GNU 所推動的通用公共授權（GNU General Public License，簡稱為 GPL），是最早也是最普及的一套開放原始碼授權條款。

漸漸地，開放原始碼已經在軟體界形成一股無法令人漠視的力量，提供自由軟體專案計畫的相關網站，也如雨後春筍般不斷出現，讓想要參與開發自由軟體的程式開發者，可以使用相當完整的資訊分享平台。除了分享平台、開放原始碼計畫的數量大幅增加外，開放原始碼授權條款的種類、數目亦不斷攀升，以最著名的自由軟體開發平台 SourceForge 網站為例，該網站所提供的開放原始碼授權條款便多達數十種。

想像中，開放原始碼授權條款的種類眾多，似乎可以提供程式開發者相當多元的選擇，實際上，太多種類的開放原始碼授權條款，只是讓程式開發者增添困擾。因為程式開發者並不會，也沒有能力逐一審視所有的開放原始碼授權條款，往往只是根據自己所

¹⁷ Richard Stallman's Personal Home Page, <http://www.stallman.org/>, (last visited Feb. 20, 2011).

知，或較為常見的幾種開放原始碼授權條款來選擇。一來確保自己知道開放原始碼授權條款的內容而受到保障，二來對想要使用原始碼的被授權人而言，也會基於相同的考量而傾向選擇較多人討論、知名度較高的開放原始碼授權條款所授權的軟體。換句話說，程式所使用的授權條款為較常見的開放原始碼授權條款時，會比較容易達到自由散布的效果，因此開放原始碼授權條款的選用亦有大者恆大的情形。

4.1.2 開放原始碼授權條款與軟體

本研究係以開放原始碼為討論標的，並以 Linux 作業系統作為主要討論對象，Linux 作業系統的原始碼採用了 GPLv2 授權條款，而作業系統屬於軟體應用中眾多類型的一小分支，因此關於軟體的法律定位如：軟體屬於著作權法的保護客體，以及軟體的可專利性等討論，本研究認為均適用於 Linux 作業系統的原始碼。

電腦軟體的程式從 *Apple Computer, Inc. v. Franklin Computer Corp.* 的判決之後，便確立其為著作權法保護的範疇，而當代各國亦皆承認電腦軟體的原始碼與目的碼¹⁸屬於著作權法的保護標的，根據 TRIPs (Trade-Related Aspects of Intellectual Property Rights) 第 10.1 條規定，電腦程式無論以原始碼或以目的碼存在，均屬於文字著作，並受到 1971 年伯恩公約的保護¹⁹。本研究係以嵌入式產品為討論對象，根據實際搭配硬體的不同，此類產品的軟體也需要對應修改原始碼，若只是取得產品內所儲存的目的碼時，即便能夠透過還原工程的幫助，得出原始碼的內容，但對於整體系統開發的實益並不大。反觀個人電腦系統，因為硬體使用了共通的規格，而能直接套用相同的目的碼，因此本研究以程式的原始碼作為主要討論對象。

軟體的可專利性過去曾是一個懸而未決的議題，但近年來軟體專利的承認已較為普遍了。相較於保護表達的著作權，專利的保護範圍更為廣泛，由於專利保護的觸角更伸向了程式背後所蘊含的思維與構想，而具有更為強大的獨占效果。

對於以軟體作為謀生工具的商業軟體公司來說，固然對於得以保障其自身權益的軟體可專利性鼓掌叫好，但對於以軟體自由為依歸的開放原始碼開發者或社群，卻認為軟體專利的核發將嚴重影響開放使碼的流通，也因此不甚贊同軟體專利的核發。

¹⁸軟體的產生過程可大致被區分為兩種格式，一為原始格式，即原始的程式碼（可肉眼判讀並理解其流程），一為執行時的程式，也稱作目的碼，市售的應用軟體為二進位格式，一般人無法閱讀此種格式而得知程式的操作流程。

¹⁹ Computer programs, whether in source or object code, shall be protected as literary works under the Berne Convention (1971). See WTO, Overview: the TRIPS Agreement, http://www.wto.org/english/tratop_e/trips_e/intel2_e.htm (last visited Jun. 4, 2010).

智慧財產權的範圍相當廣泛，智慧財產權的構想主要是利用提供短期的私人獨占保護利益來鼓勵創作人提供對大眾的智慧財產的促進，達到長期下來累積人類智慧財產的成效。軟體屬於人類智慧的結晶，因此軟體屬於智慧財產權保障對象這點是毋庸置疑的，但是開放原始碼授權條款卻顛覆了過去以私人獨占為出發點，保護創作者利益的作法，創作人藉由主動放棄部份權利的作法，來促成原始碼的流通。亦即，傳統的作法係以保護權利的方式來鼓勵創新，而開放原始碼授權條款則直接以契約的方式讓原始碼得以被分享、流通。

4.1.3 開放原始碼授權條款與網路架構

基於開放與自由精神而發展出的開放原始碼，相當無私的將軟體的內部流程予以揭露，並提供細部功能的原始碼，此種分享方式容易讓人誤以為程式開發者願意主動放棄其著作權，並混淆了開放原始碼的主張與公共領域(public domain)的思維，加上開放原始碼與網路協定同樣需要透過軟體來進行資料處理，乍看之下，容易試著將採用公共領域分享的網路架構類推至嵌入式產品的系統程式，進而將開發系統程式時對軟體堆疊 (software stack) 的規劃比擬成網路的分層架構，然而這兩者間仍有許多本質的差異，而無法直接套用。

網路協議的分層架構²⁰通常可被類比成為電話等傳輸方式的概念而被稱為層 (layer)，但是在系統軟體的開發時，軟體的分層則被稱為軟體堆疊。簡單來說，著重與外界進行溝通的網路架構，與強調內部溝通的軟體堆疊的考量是截然不同的。在網路應用中，各個分層需要與其他提供者進行溝通，藉以達到網路互連所創造出的資訊交換效益，一旦使用自我封閉的策略將喪失網路的分享本意，因此網路的分層介面涉及了與其他來源、裝置或提供者之間的溝通，需要考慮資訊的交流而必須使用相同的基本架構，也因此網路的架構若非利用公共領域的方式將喪失溝通的效果。另一方面，個別的嵌入式產品內部的系統程式卻是封閉的，嵌入式系統中的軟體採用分層開發的方式來進行之目的主要是對自身的系統內部開發分工為目的，作為私有系統使用而相當具有彈性並可任意配置。

易言之，在 TCP/IP 的網路架構中，資訊溝通管道必須是暢通的，倘若網路的架構不是公開的，而是採用個別的定義方式、或屬於特定私人時，極容易因為私益的考量而使資訊的交流受到限制。因此網路架構必須以公共領域的方式來分享，藉以確保沒有人

²⁰ 網路的分層架構所代表的除了 TCP/IP 之實體的架構外，在討論其資訊環境的內容時，亦以 physical infrastructure, logical infrastructure, 與 content 之分層架構的方式來討論，請參見：Benkler, Yochai, From Consumers to Users: Shifting the Deeper Structures of Regulation toward Sustainable Commons and User Access, 52, Fed. Comm. L. J. 561,583 (2000).

可以獨占此種溝通架構；然而嵌入式產品的系統架構卻是封閉的，在進行嵌入式產品的系統程式開發時，使用分層規劃的原因不是爲了與外界建立溝通管道，而是爲了程式的內部溝通與管理所規劃的。

需注意的是，儘管部份的嵌入式產品亦提供資料傳輸與交換功能，例如：連接至網路、USB 等介面而讀取外部資料，但此時嵌入式產品與網路、外部裝置的溝通方式是透過資料傳輸協定，也就是僅由一部分的原始碼進行與外部的溝通，並不會使嵌入式產品內部的軟體堆疊架構發生改變。

如果將嵌入式產品中的軟體開發，比喻爲蓋房子時，軟體堆疊內的架構就如同以層層搭建的方式來完成，則網路的分層架構就相當於捷運的各個路線，其資訊的流通需透過不同層的規劃來達成。爲了確保捷運車廂行經的路線是相通的，網路的分層架構必須採用一致的溝通介面，只是在車廂本體的內部設計與外部裝飾可以自由變化。

換句話說，建物的設計是根據建築師個人的偏好與需求而規劃的，建物的樓高與配置並不影響其他的建物，因此嵌入式產品內部的軟體堆疊與架構，自然會因應不同廠商的需要與定義而有所差異。另一方面，捷運路線的安排卻必須考量通道的設計，並進行整體性的規劃，否則，捷運車廂將無法移動而喪失了運輸乘客的功能，這就相當於網路分層若不一致時，便無法達成利用網路進行資訊交流的初衷。

4.2 基本分類

以開放原始碼授權條款進行軟體散布的方式，近年來形成讓傳統商業軟體公司無法漠視的力量。開放原始碼代表在軟體散布的同時，亦同時提供產生該軟體的原始碼，讓被授權人有機會獲知所使用的程式的來龍去脈，甚至讓被授權人得以依照自身的需求來對原始碼進行修改，而以一种互助互利的模式來讓軟體的功能更加精進。開放原始碼授權條款的種類相當多，而不同的開放原始碼授權條款所規範的內容，往往也代表著不同開發者的立場與想法，當前學者對於開放原始碼授權條款的分類見仁見智^{21,22}，但本研究僅說明最常見的劃分方式。

目前對開放原始碼授權條款最常見的分類方式是，依照定義開放原始碼授權條款者要求被授權人的義務深淺，而區分成爲：要求被授權人提供完整原始碼的互惠類型之開

²¹ LAWRENCE ROSEN, OPEN SOURCE LICENSING – SOFTWARE FREEDOM AND INTELLECTUAL PROPERTY LAW 69-71 (2005).

²² Gwyn Firth Murray, *Categorization of Open Source Licenses: More Than Just Semantics*, 26 THE COMPUTER & INTERNET LAWYER 1, 1-11(2009).

放原始碼授權條款 (Reciprocal Licenses)、僅要求被授權人提供部份原始碼之折衷型的開放原始碼授權條款，以及不要求被授權人提供原始碼之學術類型的開放原始碼授權條款 (Academic Licenses) 三種。其中互惠類型之開放原始碼授權條款，主要由社群所提出，而折衷型的開放原始碼授權條款，係由商業公司所推動，學術類型的開放原始碼授權條款則由學術機構所提出。

需注意的是，無論開放原始碼授權條款所屬的類型為何，其授權對象並不會依據提出授權條款的組織為何而受到影響。以學術類型的開放原始碼授權條款為例，該種類之開放原始碼授權條款係由學術機構所提出，但該些開放原始碼授權條款，並不會限制開放原始碼僅能作為學術用途。

上述的開放原始碼授權的分類方式就如頻譜的分布，在頻譜的最左邊的是對自由要求極大化的互惠型開放原始碼授權條款，這類型的開放原始碼授權條款要求開放原始碼的衍生著作，必須依照相同的授權條款進行授權，也就是必須提供衍生著作的原始碼，因此被稱為具有 Copyleft (著佐權²³) 效力；位於頻譜最右邊的則是學術類型的開放原始碼授權條款，這類型的開放原始碼授權條款對於被授權人的義務規定相當簡單，並不限定衍生著作一定要採用同樣的開放原始碼授權條款，也不限定被授權人必須提供衍生著作的原始碼，而僅對公開發表日期、著作權人等標示加以限定，是最接近公共領域概念的開放原始碼授權條款；在頻譜中間的則是折衷型的開放原始碼授權條款，這類型的開放原始碼授權條款並不會限制衍生著作必須提供原始碼，而能以模組化等方式劃分程式原始碼的不同部份，並針對不同的原始碼來源使用不同的授權條款。

4.2.1 互惠型開放原始碼授權條款

互惠型的開放原始碼授權條款最大的特色是「相同方式分享」，由 GNU 所制定的各版本的 GPL、LGPL 授權條款是互惠型開放原始碼授權條款的代表，互惠型的開放原始碼授權條款最大的特色便是著佐權的影響範圍最廣，藉由授權契約的規範，讓程式的提供者與被授權人均享有對原始碼進行使用、修改、重製與發展衍生著作的權利，且被授權人後續在重製與散布原始碼的衍生著作的同時，這些權利亦會透過相同的條件傳遞。易言之，著佐權的規定會讓被開放原始碼的授權人在散布程式時，所需履行的責任較多。

附帶一提的是，儘管 GNU 在 2007 年發行了 GPL 第三版 (GPLv3 授權條款)，但新版的 GPLv3 授權條款與 GPLv2 授權條款並不相容，多數使用 GPLv2 授權條款的開放原

²³巫坤霖，由著作權到著佐權-從著作權法及競爭法交錯之角度分析電腦程式著作授權契約，台大法研所碩士論文，(2009)。

始碼仍選擇繼續採用 GPLv2 授權條款，也因此目前 GPLv2 授權條款仍居於開放原始碼授權條款中的主流地位。再者，本研究所探討的 Linux 作業系統仍維持使用 GPLv2 授權條款，因此若本研究中未特別標明時，GPL 授權條款係指 GPLv2 授權條款。

4.2.2 折衷型開放原始碼授權條款

折衷型的開放原始碼授權條款中，最典型的代表是 Mozilla Public License(簡稱為 MPL 授權條款)，這種類型的開放原始碼授權條款提供被授權人相對較大的使用權利，與相對較少的義務規範，最明顯的就是著佐權的範圍較小，亦即，會受到著佐權影響的原始碼，僅限於最初以開放原始碼授權條款進行授權的部份，被授權人自行開發的原始碼若能與最初的開放原始碼間進行區隔時，被授權人自行新增的程式便不會受到感染。簡言之，廠商的程式來源若採用的是此種類型的開放原始碼授權條款時，廠商需提供原始碼的義務便較互惠類型的開放原始碼授權條款輕許多。

關於 MPL 授權條款的規定，可被簡單詮釋如下，這句話呈現了折衷類型的開放原始碼授權條款的主要精神²⁴：

原文：

If you create and distribute a Modification to one of the files containing Original Code or previous Modifications, or create and distribute a new file containing Original Code or previous Modifications, those files must be released as modifications under the same MPL license.

翻譯：

若您對原始程式或先前的修改或建立且散布一個檔案包含原始程式或先前的修改，這些檔案與修改必須同樣使用 MPL 授權條款來公佈。

其他的折衷型開放原始碼授權條款，有的是以模組作為授權條款範圍的選定，但無論是以檔案或以模組作為程式劃分的依據，均可看出這種類型的開放原始碼授權條款，要求原有使用開放原始碼授權條款的原始碼，需要繼續維持原本的授權條款進行授權，由於 MPL 授權條款的著佐權效力不及於新增的程式，因此被授權人自己新增的程式並不需要採用相同的開放原始碼授權條款。

舉例來說，當開發者（甲）所發展的軟體包含了兩個檔案，其中 a 檔案係為以 MPL

²⁴ See ROSEN, *supra* note 21, at 69-71.

授權條款進行授權的開放原始碼，而 b 檔案為甲自行開發的，且甲對 a 檔案進行修改而得到 a1 檔案，則 a1 檔案（相當於 a 檔案的衍生著作）需要採用 MPL 授權條款，而 b 檔案則不需要被限制採用 MPL 授權條款。

4.2.3 學術型開放原始碼授權條款

這類型的開放原始碼授權條款的起源來自於學術界，但不限定使用者僅得為學術界方能使用，此種類型的開放原始碼授權條款中，又以加州大學柏克萊分校所發展出的 BSD 授權條款（Berkeley Software Distribution，簡稱為 BSD）為代表，這類型的授權條款對於程式的被授權人是最寬容的。

以 BSD 授權條款為例，BSD 授權條款只要求被授權人附上 BSD 授權條款的原文與聲明開發者的著作權資料即可。亦即，BSD 授權條款給予被授權人的自由度很大，也不使用著佐權的規定，因此原始碼採用 BSD 授權條款時，被授權人將衍生著作傳遞給後手時，該些衍生著作並不需要採用 BSD 授權條款，而允許他人使用、改作和散布程式的衍生著作（包含程式的原始碼與目的碼）。這類型的開放原始碼授權條款要求被授權人保留其名聲，而這也和此授權條款的起源為學術單位有密不可分的關係。

BSD 授權條款認為，軟體就該提供給別人自由使用，不應該限制軟體的使用權限，BSD 授權條款強調的是全面的自由，既不限制被授權人的複製與散布自由，在被授權人複製與散布程式時，亦不特別強求被授權人需要提供原始碼。

由於學術類型的開放原始碼授權條款，並未限制衍生著作應採用的授權方式，也就是被授權人可以選擇自己偏好的授權方式，此種相當具有彈性作法，對企業而言可以不用擔心著佐權的影響，也讓廠商使用開放原始碼開發產品時，沒有後顧之憂而相當的便利。當嵌入式產品的開發廠商取得的開放原始碼係以學術類型的授權條款進行授權時，代表廠商可以利用現成的原始碼，卻不需要提供自己修改後的衍生著作，此種開放原始碼授權條款的規定對廠商而言，相當於只需享受權利但義務相對較少的規定，也因此相當受廠商的歡迎。

4.2.4 相容性

簡單來說，前述對於開放原始碼授權條款的分類方式固然係依據規範內容來分來，但進一步探討規範內容時，可以發現該些規範內容係由不同提出者根據其立場而訂定的。例如：互惠型的開放原始碼授權條款，代表了以授權條款迫使全部的被授權人進行原始碼共享為主要訴求的支持者；折衷型開放原始碼授權條款的擁護者則以商業公司為

主，其目的是希望能藉由開放原始碼促進商業化的程式更進步，但在商業競爭的考量下，他們對於原始碼的開放態度也較為保留；至於學術型開放原始碼授權條款因為起源於大學或研究機構，因此其分享原始碼的初衷係著眼於學術地位的，因此僅要求被授權人必須標示其來源，並不堅持原始碼需要維持其共享的特性。

易言之，訂定開放原始碼授權條款的團體或個人的立場，往往左右了開放原始碼授權條款的內含，在不同的考量下，各派的主張也彼此衝突，映射至該些團體所分別規範的開放原始碼授權條款時，也連帶使開放原始碼授權條款的相容性產生衝突與矛盾。

開放原始碼授權條款之間的相容，指的是根據不同的開放原始碼授權條款所授權的原始碼是否可以被合併使用，因此開放原始碼授權條款彼此之間是否相容必須兩兩進行比對，若是程式開發者在開發程式的時候，不同來源的原始碼所使用的授權條款的規範中，彼此不發生互相衝突時，便稱為該些授權條款彼此相容；至於程式開發者在修改程式時，若使用了其他授權條款的原始碼對原本的原始碼進行修改時，若新舊程式所使用的開放原始碼授權條款不發生衝突便具有相容性，因此授權條款間若具有相容性時，可使原始碼的整合與流通更為便利²⁵。

換句話說，若是一起使用了利用 A 授權條款的 a 程式以及利用 B 授權條款的 b 程式而衍生 c 程式時，a 授權條款與 b 授權條款的規定如果不一致時，c 程式是否能單獨採用 a 授權條款，或單獨採用 b 授權條款？以下便將上述的三種開放原始碼授權條款的類型中，最具代表性的 GPLv2 授權條款、MPL 授權條款與 BSD 授權條款作為舉例，將該些開放原始碼授權條款兩兩進行比對，藉以探討各種類型之開放原始碼授權條款的相容性²⁶。

4.2.4.1 GPL 與 MPL

互惠型開放原始碼授權條款的特徵是，保障衍生著作仍使用相同的開放原始碼授權條款進行授權²⁷，代表這種類型的開放原始碼授權條款，要求程式的衍生著作須繼受互惠型授權條款的授權內容（如：GPLv2 授權條款），也就是衍生著作也必須提供原始碼，導致所有的程式因為此種感染性而變成使用相同的互惠型授權條款，而 MPL 授權條款卻允許程式的原始碼可以根據來源的不同而使用不同的授權條款。

²⁵李界昇，開放原始碼模式下的法律糾紛與風險-從 SCO v. IBM 案出發，交大科法所碩士論文，頁 56 (2004)。

²⁶由於各類型的開放原始碼授權條款亦涵蓋許多不同的授權條款內容，因此僅以各類型中最具代表性的 GPL、BSD 及 MPL 進行相容性的討論。

²⁷馮震宇，「開放原始碼授權契約之法律問題與挑戰」，全國律師，第 9 卷第 1 期，頁 3-17 (2005)。

採用 GPLv2 授權條款的原始碼並不會因為與以 MPL 授權條款所授權的原始碼共用而改變其授權為 MPL 授權，而原本以 MPL 授權條款所授權的原始碼也不因之而變成以 GPLv2 授權，加上兩者規範的內容有所衝突，即，GPLv2 授權條款將限縮 MPL 授權條款的範圍，因此 GPLv2 授權條款與 MPL 授權條款並不相容。

4.2.4.2 MPL 與 BSD

學術型開放原始碼授權條款也被稱為是保障衍生著作仍是開放原始碼的類型，以及准許被授權人得自由利用該軟體之權利的類型²⁸，因此當折衷型的 MPL 授權條款與學術型的 BSD 授權條款一起被使用時，MPL 授權條款尊重 BSD 授權條款的規範內容，而不會限制 BSD 授權條款的視用範圍，而 BSD 授權條款亦不會對他種授權條款產生影響。

易言之，採用 MPL 授權條款的原始碼，與採用 BSD 授權條款的原始碼搭配使用時，原始碼授權方式並不會產生變化，也就是說，整體的程式原始碼中，原本採用 MPL 授權條款的原始碼仍維持以 MPL 授權，原本採用 BSD 授權條款仍維持以 BSD 授權條款進行授權，而不會因為原始碼共用而使授權條款的規範產生衝突。

4.2.4.3 BSD 與 GPL

由於 GPLv2 授權條款具有著佐權的效力，導致原本利用 BSD 授權條款授權的程式，會因為與採用 GPLv2 授權條款的程式一起使用，而被 GPLv2 授權條款同化的現象。另一方面，由於 BSD 授權條款並不限定衍生著作所使用的授權條款，也不具有著佐權的特性，因此，原本以 GPLv2 授權條款進行軟體授權的程式，卻不會有被 BSD 授權條款同化的情形。換言之，BSD 授權條款與 GPLv2 授權條款間，形成了單行道的現象，即，GPLv2 授權條款會感染原本以 BSD 授權條款進行授權的程式，但 BSD 授權條款卻不會感染以 GPLv2 授權條款進行授權的程式。

換句話說，若軟體程式同時包含 GPLv2 授權條款與 BSD 授權條款的程式，便會因為 GPLv2 授權條款的感染性，使得所有的程式都一併變成了使用 GPLv2 授權條款。對奉軟體自由為圭臬的程式開發者而言，採用互惠型的開放原始碼授權條款，可以迫使開放原始碼的衍生著作自由流通，但是不同的程式開發者對於原始碼的開放程度與使用自由，卻可能有不同的看法。

²⁸同前註。

舉例來說，選擇 BSD 授權條款的程式開發者在分享程式原始碼後，如果其他人在開發程式時，除了使用他的原始碼外，自己的原始碼採用 GPLv2 授權條款，將使原作者的原始碼受到 GPLv2 授權條款的感染，也必須由原本的 BSD 授權方式改為 GPLv2 授權方式。換言之，原作者的原始碼被迫改採非由原作者所選擇的開放原始碼授權，甚至可能會覺得自身的權益受損。

事實上，部份支持學術類型的開放原始碼授權條款的擁護者認為，互惠型的開放原始碼授權條款本身也是一種限制，因為互惠型開放原始碼授權條款限制了其他人的自由，強制被授權人在散布程式時，必須採用相同的開放原始碼授權條款並提供程式的原始碼，只是限制的手段與商業軟體要求不能散布的作法恰巧相反。

雖然使用 GPLv2 授權條款的自由軟體專案數量，一直遙遙領先採用其他開放原始碼授權條款的專案，但對於以商業利益作為最大考量的企業而言，在擔心研發機密外流的考量下，便無法坦然接受 GPLv2 授權條款以分享為依歸的作法。而這也是一般人對於開放原始碼普遍存在誤解的原因，認為開放原始碼授權條款存在的目的是為了與商業公司唱反調。

由於開放原始碼授權條款的種類相當多，並非每一種開放原始碼授權條款都會規定被授權人需要提供原始碼，廠商還是可以使用以 BSD 授權條款、MPL 授權條款的開放原始碼，只要在遵守該些開放原始碼授權條款的規範下，盡到標示來源或提供原始著作的原始碼之責任即可，亦即，對於講求實際獲益的廠商而言，仍然可以在保護研發機密的情況下，善加利用開放原始碼的資源。

4.3 法律定位

由於本研究係以 Embedded Linux 作為討論的核心，而 Linux 作業系統為眾多軟體應用中的一小分支。換句話說，Linux 作業系統具有軟體的特性，因此本研究將焦點關注在開放原始碼所衍生的法律問題，而非軟體的法律定位。

開放原始碼授權條款在問世後，引發了相當多爭議。儘管最初開放原始碼授權條款的爭訟大多以著作權爭議為主²⁹，但是認真細究開放原始碼授權條款的規定時，可以發現，開放原始碼授權條款的規範內容，可能涉及民法的契約、著作權法、專利、競爭法、

²⁹ 例如：由 SCO Group 對 IBM、Daimler-Chrysler 等廠商提起的幾件爭訟案件，請參見：李科逸，「自由軟體國際間智財權重要爭訟及趨勢之淺析-著作權訴訟爭議漸息，專利權侵權疑慮頻生」，智慧財產權月刊，第 76 期，頁 5-27（2005）。

營業秘密³⁰等多種法律議題³¹，且每一個主題皆足以利用專論來探究。爲了不使討論主題過於發散，本研究不擬逐一討論開放原始碼授權條款在各該領域之定位、爭議與因應策略，僅將開放原始碼授權條款視爲著作權授權契約³²，並簡要說明開放原始碼授權條款與著作權、專利、競爭法的關係。

4.3.1 開放原始碼授權條款與著作權

在探討開放原始碼與著作權的關係時，經常會涉及開放原始碼的原創性，與誰才是開放原始碼的著作權擁有者等議題，然而本研究將不討論該些議題，而著重於開放原始碼授權條款的規定。開放原始碼授權條款係以原始碼爲核心，而原始碼又爲著作權的保護標的，因此開放原始碼的法律問題，經常著重在開放原始碼授權條款的規定與著作權之間的是否一致，或是否產生矛盾³³。

簡言之，本研究在此將開放原始碼授權條款視爲，以電腦程式著作，作爲著作權法保護標的前提，由電腦程式的著作權人在發行其程式時，將程式原始碼的散布與重製等情形，附帶條件並予以敘明的著作權授權契約。

4.3.1.1 著佐權

由於著佐權打破了傳統著作權關於使用自由的限制作法，因此著佐權常常被視爲與傳統的著作權相對立。事實上，著佐權的運作係建立在著作權的基礎上³⁴，其係以著作權法所授予著作權人的法律權利爲依據，賦予著作權人要求被授權人遵守相關規定的一種規定。

正因爲如此，具有著佐權性質的開放原始碼授權條款，應該被視爲是一份要求提供原始碼，提供共享自由的著作權聲明文件。易言之，具有著佐權性質的開放原始碼授權條款的起因是，爲自由軟體的推動者認爲傳統著作權法的保障不符合其需求，而利用授

³⁰ Alberto Pianon, *Trade Secret Vs. Open Source: And The Winner Is ...*, 1 ERASMUS L. & ECON. REV. 47 (2004).

³¹ Tomasz Rychlicki, *GPLv3: New Software Licence and New Axiology of Intellectual Property Law*, 30 EUROPEAN INTELL. PROP. REV., 232 (2008).

³² 楊千、廖先志、陳鍾誠，「從我國法制論 GPL 授權契約-兼論第 3 版草案之適用」，台灣科技法律與政策論叢，第 3 卷第 3 期，頁 121-150 (2006)。

³³ 舉例來說，根據 GPLv2 授權條款的規定，一旦被授權人使用了以 GPLv2 授權條款進行授權的原始碼，無論所使用的原始碼之多寡，被授權人的程式同樣受到 GPLv2 授權條款的限制，並不能因爲主張著作權法所保障的合理使用而免除 GPLv2 授權條款的要求，此時 GPLv2 授權條款與著作權法的規定便有所出入，參見：Douglas D. McGhee, *Free and Open Source Software Licenses: Benefits, Risks, and Steps Toward Ensuring Compliance*, 19(11) Intellectual Property & Technology Law Journal, 5,9 (2007).

³⁴ LAWRENCE LESSIG, *FREE CULTURE-HOW BIG MEDIA USES TECHNOLOGY AND THE LAW TO LOCK DOWN CULTURE AND CONTROL CREATIVITY* 264-265 (2004).

權條款作為改善原本著作權法規範的缺失之手段，因此應被視為一種較為特殊類型的著作權聲明，儘管在開放原始碼授權條款中，著作權人放棄了部份原本著作權保障的權利，但仍其保留之權利仍屬於著作權的範疇。

4.3.1.2 數位權利管理

數位權利管理（Digital Rights Management/Digital Restrictions Management，簡稱為 DRM）通常被商業廠商用來控管其著作被利用的方式，主要目的在限制他人對其著作之使用³⁵，常見於影音產品或一般的商業軟體中，數位權利管理通常被廠商用來避免他人在未經授權的情況下取得資料內容，而這恰恰與開放原始碼授權條款所強調的自由背道而馳，關於這部份的討論請參見下一章，針對科技保護措施與 GPLv2 授權條款所引起的爭議之相關討論。

4.3.2 開放原始碼授權條款與專利

相較於著作權法，開放原始碼授權條款與專利權的關係更為緊張。專利權制度係以保護的手段，讓發明人得以在一段期間內享有排他的權利，在該段期間內，發明人得以享有商業上的利益，藉此來鼓勵專利所有權人分享其發明。換言之，專利權所強調的排他權，與自由軟體的擁護者所提倡的分享理念是背道而馳的。為此，目前有不少開放原始碼授權條款明定，一旦軟體專利被使用在開放原始碼中，則視同專利權人願意分享其專利內容而不得主張其專利權³⁶。

與著作權用來保護表達而不保護思想不同的是，專利權更進一步的保護了以專利範圍所涵蓋的構想，因此，即便原始碼的內容不完全相同，但用來達成手段相類似的情況下，仍有可能形成專利權的侵害。換句話說，若有一商業軟體廠商開發了軟體 A，其原始碼為 a 程式，在著作權的保障下，如果有其他的軟體廠商或個人提供了類似與軟體 A 的功能相類似的軟體 B，且軟體 B 的原始碼為自行開發而非抄襲軟體 A 的 a 程式，則 a 程式的著作權並未受到侵犯。然而，若是軟體廠商就 a 程式的相關流程，利用方法專利等方式進行申請並獲准專利，則軟體 B 的作法便可能被專利範圍所包含，導致專利侵

³⁵ 著作權筆記，何謂數位權利管理：<http://www.copyrightnote.org/cnote/bbs.php?board=3&act=read&id=530>（最後點閱時間：2010 年 5 月 5 日）。

³⁶ 依據 GPLv3 授權條款的第十條第三項所規定“*You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.*”參見：GNU General Public License, <http://www.gnu.org/licenses/gpl.html>（最後點閱時間：2011 年 3 月 9 日）。

權。

正因為專利具有相當大的殺傷力，也常被商業公司作為打擊對手時的攻擊武器，因此這些以軟體為牟利工具的公司也會以大量申請軟體專利的方式，打擊開放原始碼運動。專利權的存在，固然鼓勵了發明人分享其想法，但對於其他大眾若要加以使用時的負面影響卻是必然的，因為專利是利用商業上的利益來鼓勵發明人分享其發明，而這與自由軟體崇尚自由使用的理念是背道而馳的，因此，Stallman 極力反對軟體專利的存在，他認為軟體專利權人所被賦予的權利妨礙了資訊的流通與分享，因此有許多開放原始碼的授權契約新增有關專利授權的條款，甚至明定，一旦軟體專利被使用在利用相關的開放原始碼時，則專利權人不得據此而提起專利侵權的訴訟³⁷。

儘管透過開放原始碼授權條款的規範，要求釋出開放原始碼的授權人，不得因此而獲得專利權上的利益所賦予授權人的義務，可以讓軟體專利的使用情形受到限制，藉以保護被授權人不會因為使用開放原始碼而導致侵犯專利的情形。但這些條款僅對於專利所有權人將其專利內容加諸於開放原始碼授權條款的程式中有效，對於未擁有專利權但撰寫出可能涉及他人專利權的開放原始碼程式的情形，仍可能造成專利侵害的情形。

2004 年 8 月，開放原始碼風險管理公司(Open Source Risk Management, 簡稱為 OSRM) 委託公共專利基金會(Patent Foundation)所進行的調查便指出 Linux 作業系統的原始碼可能含有引發專利侵權糾紛的美國專利共有 283 項³⁸。這些專利的擁有者雖然包含了 IBM、HP 等被視為是開放原始碼運動盟友的企業，但其中的部份專利的擁有者，卻是讓社群相當感冒的商業軟體龍頭微軟。

開放原始碼社群除了與商業軟體公司就軟體專利的應否存在而發生激辯外，開放原始碼授權條款與專利權還存在許多問題。舉例來說，Linux 作業系統的原始碼是否可能夾帶著侵犯他人專利權的原始碼？再者，若是開放原始碼的確侵犯了他人的專利權，究竟是專利權人自願提供於原始碼中，抑或在不知情的情況下遭到侵犯？倘若開放原始碼所使用的軟體專利，未經專利權人的授權時，又該如何彌補專利權人的損失？應用在開放原始碼的專利權是否真的具有有效性？身為開放原始碼的被授權人，又該如何避免這些專利權的所有人在日後提出告訴？種種問題都可以看出開放原始碼授權條款與專利權之間的糾葛相當複雜。惟本研究受限於篇幅，對於該等議題並未多作討論。

³⁷ 如：GPLv3 授權條款。

³⁸ OSRM Position Paper: Mitigating Linux Patent Risk, http://www.osriskmanagement.com/pdf_articles/linuxpatentpaper.pdf (last visited, Mar. 9, 2011).

4.3.3 開放原始碼授權條款與競爭法

競爭法的目的是爲了要維護自由與公平競爭的市場環境，通常競爭法主要與商業團體、活動相關，由於電腦軟體在開發時，研發的成本遠高於重製與銷售的成本，過去電腦軟體與競爭法有所關聯時，往往是商業軟體大廠利用商業手段造而造成的爭議，例如微軟的瀏覽器與 Netscape 之爭，以及微軟在歐盟所面臨的相關訴訟等。

競爭法的內涵包含限制競爭與防止不正競爭兩種，前者是爲了維護市場的機能，而對具有市場影響力的寡占或獨占行爲加以限制，而後者則是用來規範違反競爭倫理而妨礙市場公平競爭的行爲。換句話說，限制競爭的目的是爲了避免市場的壟斷，而造成市場機制的失靈，而防止不正競爭的目的則是導正交易秩序，兩者之間相輔相成，雖有部分的內容有所重疊，但用來保護的立論基礎則不同。

乍看之下，推廣開放原始碼運動的社群並非以商業行爲爲目的而與競爭法毫不相干，但 GPLv2 授權條款曾經在美國 Wallace³⁹ 以及在德國的 Skype⁴⁰ 的兩個訴訟中被提出，認爲 GPLv2 授權條款的規範內容違反了競爭法的規定。

在美國的 Wallace 控訴自由軟體基金會 (Free Software Foundation, 簡稱 FSF) 一案中，Wallace 欲使用 BSD 授權條款作爲自己所開發的程式所用，卻因爲他所開發的程式使用了 Linux 作業系統的原始碼，導致其程式被 GPLv2 授權條款所感染，連帶的其所撰寫的原始碼也被迫使用 GPLv2 授權條款，因此 Wallace 便以 GPLv2 授權條款透過將 Linux 定價爲 0 的手段，而以掠奪性定價 (predatory pricing claim) 的方式排擠競爭對手爲由，控告大力投入 Linux 作業系統開發的 IBM、Red Hat 與 Novell 等公司。Wallace 的控訴在美國的判決得到敗訴，承審法官在判決中對 GPLv2 授權條款表達了支持的立場，並認爲自由軟體授權條款所推崇就是自由與開放，因此即使在授權條款中強制要求後手在散布程式時必須要提供原始碼並不得收取授權費用，但對於市場秩序所採取的亦以自由 (完全競爭) 爲要求，其實是有助於市場的自由競爭的，因此 Wallace 所提出的不正競爭指控是不成立的。

Skype 與 Welte 在德國的訴訟中，原本提出了上訴，在其上訴理由中，宣稱 GPLv2 授權條款違反了德國的反壟斷法，並因此認爲 GPLv2 授權條款在德國應屬無效。然而上訴法院認爲，即使 GPLv2 授權條款在德國被認定爲無效，Skype 仍然無權使用採用該授

³⁹ 7th Circuit Holds GPL and Open Source Software Do Not Violate Antitrust Law, <http://www.techlawjournal.com/alert/2006/11/10.asp> (last visited Mar. 6, 2011).

⁴⁰ Victory: Skype Withdraws Appeals Case, Judgement from Lower Court Accepted, http://laforge.gnumonks.org/weblog/2008/05/08/#20080508-olg_muenchen-skype (last visited Mar. 6, 2011).

權條款的軟體，最後 Skype 便撤銷了該上訴。

4.4 小結

儘管同樣以利他主義為出發點，但開放原始碼授權條款卻與公共領域不同，公共領域的作者放棄了自身所有的權利，將其著作全然的分享給大眾，而開放原始碼的作者僅放棄部份的權利，並要求其被授權人遵循一定的規範，方能取得該些權利，其分享是附條件且出於特殊動機。另一方面，開放原始碼授權條款所採用的作法，又與傳統著作權的思維格格不入，這是因為開放原始碼授權條款所欲保障的使用與散布自由等訴求，與傳統智慧財產權以保障私有權利之構想截然不同，如何妥善解決這些矛盾和問題，則是產業界與法律界必須共同面對的挑戰。

自由軟體運動推行至今，背後所蘊含的著作權理念亦形成一個相當成功的著作分享機制，而類似的觀念也逐漸被應用到其他種類著作的分享，例如 Wikipedia、Creative Commons 等。事實上，此股以分享促進發展的構想，以及社群的開發力量，已經讓身為軟體帝國的微軟無法漠視，讓微軟感受到不小的壓力。為了順應時勢，微軟目前已經推出兩套受到 OSI 認證的開放原始碼授權條款^{41,42}，這也代表微軟從大肆抨擊自由軟體，認為這種模式所開發的軟體品質堪虞的態度，逐漸轉變為對正視開放原始碼議題，甚至放下身段的作法。

開放原始碼授權條款的類型相當多，不同類型的開放原始碼授權條款對於商業使用並不盡然都抱持對立的態度，因此對嵌入式系統開發廠商而言，大可不必聞開放原始碼授權條款而色變。相反地，廠商應該摒棄成見，試著與開放原始碼授權條款和平共處，亦應放棄鴛鴦將頭埋在沙堆而粉飾太平的心態，以更積極的態度瞭解開放原始碼授權條款背後的意含，以及使用開放原始碼時所需注意的相關法律問題。

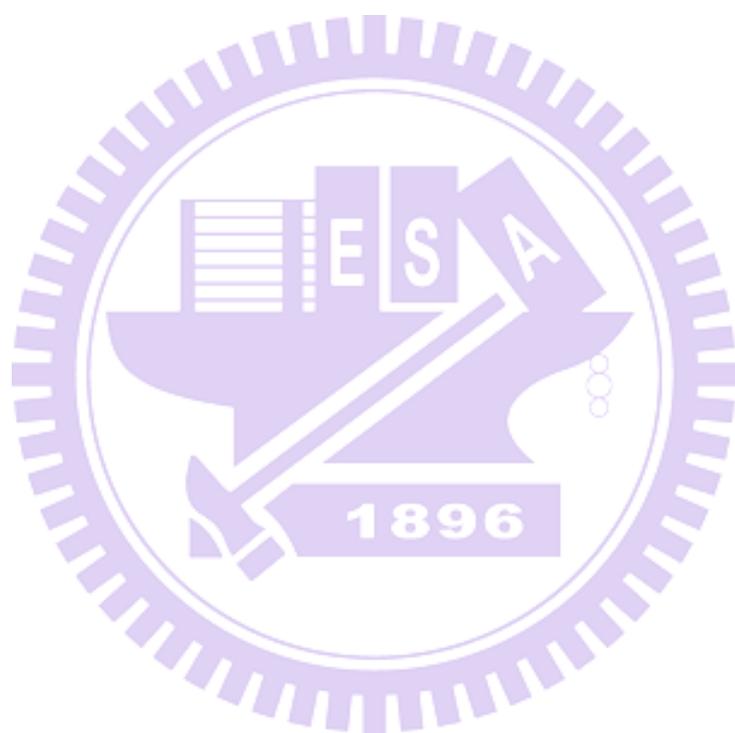
附帶一提的是，即便廠商選擇使用的開放原始碼採用的是約束較少的折衷類型或學術類型之授權條款，讓廠商不需被迫提供完整的原始碼。但該些開放原始碼授權條款的規範內容，如揭示開放原始碼的原始創作者的名稱、標示原始碼來源等規定，被授權人亦應遵守，否則仍然可能違反該些開放原始碼授權條款的規定。

開放原始碼授權條款的法律性質，以及與他種法律之間的關係，可參考相關文獻的

⁴¹ Microsoft Reciprocal License (Ms-RL)以及 Microsoft Public License (Ms-PL)，其中前者比較類似 MPL 授權條款，後者比較類似 BSD 授權條款。

⁴² Microsoft, *Open Source Licenses*, <http://www.microsoft.com/opensource/licenses.mspx> (last visited Jan. 17, 2010).

討論⁴³。本研究僅簡要論述開放原始碼可能產生的法律議題，並將討論的重心放在將開放原始碼應用於嵌入式產品時的特殊考量。



⁴³ 康雲龍，論開放原始碼軟體對智慧財產權理論之影響---以著作權法為中心，銘傳法研所碩士論文，(2007)。

林誠夏，自由軟體產業推動之反思-以加拿大資訊產業發展政策為比較觀點，台大國發所碩士論文，(2007)。

巫坤霖，前揭註 23。

第五章 Embedded Linux 的授權範圍

5.1 Embedded Linux 與 GPLv2 授權條款

Torvalds 在 1990 年代開發 Linux 作業系統的原始碼的時候，因為贊同 GNU 的理念，也希望自己所開發的程式能夠藉由與他人的交流而更進步，因此選擇了 GPLv2 授權條款。時至今日，參與 Linux 作業系統的開發者已經不計其數，讓 Linux 作業系統的開發越見成熟，進而讓 Linux 作業系統成為許多嵌入式產品廠商的首選。

根據第四章的說明，GPLv2 授權條款為了保障所有利用此類型授權條款的原始碼能夠被自由使用，因而限定根據 GPLv2 授權條款的程式為基礎而開發的原始碼，皆必須使用 GPLv2 授權條款，也因此，使用 GPLv2 授權條款的原始碼就如同具有感染性一般，會將原始碼必須強制授權的特性傳遞到任何相關的程式⁴⁴。

換句話說，一旦有一個 a 程式使用了 GPL 作為其授權條款，不但讓使用 a 程式的 b 程式成為以 GPLv2 授權方式進行授權，連帶的，使用 b 程式的 c 程式也會因為 b 程式的 GPLv2 授權條款之效力而一併成為 GPLv2 授權條款的適用對象，就如同毒樹果實般，讓開枝散葉的原始碼均受到感染。亦即，GPLv2 的授權會以漸進式的方式，將授權規定傳遞給相關的原始碼。亦即，由於 GPLv2 授權條款具有強烈的著佐權特性，一旦程式選擇了 GPLv2 授權條款，將造成感染效力一傳十、十傳百的效果，讓所有的程式原始碼，都因為受到 GPLv2 授權條款的規範而需提供給後手。

正因為 Linux 作業系統選用的是 GPLv2 授權條款，因此 Linux 作業系統的被授權人不但可以單純的使用 Linux 作業系統，還可以取得並對 Linux 原始碼進行改作，但在 GPLv2 的規範下，被授權人對於 Linux 原始碼的改作結果必須提供給其他人，並維持以 GPLv2 作為授權條款。基於 GPLv2 授權條款特有的強烈感染力，一旦選擇使用 Linux 作業系統，在作業系統對下與驅動層相連，讓驅動程式受到感染而必須提供給下游的被授權人，對上則導致應用程式層，甚至使用者介面層也同樣受到感染的情況下，似乎將導致全部的系統程式都需要依照 GPLv2 授權條款，提供給程式所散布的對象，也就是嵌入式產品的一般使用者（消費者）、購買者。

⁴⁴ The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. See GNU Operating System, <http://www.gnu.org/licenses/gpl-2.0.html> (last visited Mar. 22, 2010).

綜上所述，使用了 GPLv2 授權條款的 Linux 作業系統，對於系統程式的感染效力非常驚人，嵌入式產品開發廠商也對這樣的認定方式感到相當不合理，採用這種解讀方式時，即便作業系統的原始碼僅佔了全體系統程式之原始碼的百分之十，但是其他各層的原始碼也需一併提供給後手，在所得之助益有限、所付出之對價相對無止盡的考量下，廠商的反彈也是可以預見的。

另一方面，就嵌入式產品的系統程式來說，由於程式是透過硬體燒錄目的碼的方式而存在於嵌入式產品中，因此嵌入式產品的販售便代表程式的散布。對嵌入式產品的一般使用者而言，關心的是產品的功能是否運作正常，也就是被燒錄於非揮發性記憶裝置（如：快閃記憶體）內的軟體，是否能夠正確的對嵌入式產品的硬體週邊進行控制。亦即，絕大多數的使用者對於嵌入式產品本身究竟以何種方式運作、程式如何進行控制並不在乎，因此廠商是否提供軟體的原始碼，以及廠商的系統程式係以何種方式來架構等問題，並不會影響一般使用者對嵌入式產品的選擇。

以智慧型手機的選擇為例，一般使用者並不會因為手機平台是否為開放原始碼而選擇，而是根據自己的需求而選擇合適的手機，例如：偏好多媒體娛樂功能的使用者，可能選擇使用 iPhone，而著眼於事務用途的使用者則可能選擇 Android 平台的手機等。因此，本研究認為，開放原始碼授權條款的起源係以工作站、個人電腦等平台為主，在該些平台上，軟體應用的能見度較高，而嵌入式產品之本質既然與該些平台不同，便不應該以相同的授權條款內容來規範。

也就是說，開放原始碼授權條款的規範內容，應該針對應用平台的特性而調整，否則以 GPLv2 授權條款的強大感染力，只會讓嵌入式產品的開發廠商感到畏懼，而採取保守、封閉的態度，當嵌入式產品的開發廠商不願意正視開放原始碼的議題時，開放原始碼的推展也會連帶受到影響。

5.2 GPLv2 授權條款的規範

以下的說明將 GPLv2 授權條款的內容區分為三類，首先是屬於背景介紹的前言，以及說明使用 Linux 作業系統的嵌入式產品，為何會在廠商未簽署 GPLv2 授權條款的情況下，受到 GPLv2 條款的規範（第 5 條）；第二種類型是特別就廠商在開發嵌入式產品時的考量而特別容易違反的幾個 GPLv2 授權條款之條款內容；第三類型的條款則是與嵌入式產品本身特性較無絕對相關，而與一般個人電腦應用之違反情形較為類似的條款。

根據條款類型的不同，在討論第一種類型與第三種類型的條款時，僅以歸納整理的

方式說明，而與嵌入式產品的開發息息相關之第二種類型的條款，則會引用條款的內容並進行翻譯，以及對照必較該些條款的規定，與嵌入式產品廠商實際在使用開放原始碼時的應用情形。附帶一提的是，GPLv2 授權條款的原文⁴⁵與中文翻譯⁴⁶可在網路上取得，本研究在處並不全文引述，僅節錄其中與嵌入式產品特別相關的條款。

5.2.1 GPLv2 前言

GPLv2 授權條款的前言說明了整份 GPLv2 授權條款的規範方向，與主張自由使用軟體的思維，雖然前言並不屬於 GPLv2 授權條款的規範內容，也不至於影響授權條款的效力，但 GPLv2 授權條款的前言因為使用了較為強烈的用語，而易令被授權人感到畏懼⁴⁷。

在前言中，首先區隔了開放原始碼授權條款與專屬授權條款最大的差別便是，專屬授權條款被用來剝奪被授權人共享和修改軟體的自由，而 GPLv2 授權條款卻被用來保證被授權人共享和修改軟體的自由。另外，前言也大致說明了被授權人在使用 GPLv2 授權條款所能享有的權利，即，利用著作權來保護軟體，以及藉由 GPLv2 授權條款的規定，讓程式開發者得以自由的散布軟體，並可以自由決定是否要對此服務收取費用；確保被授權人得以取得並修改開放原始碼，或應用至其他原始碼中；以及程式提供者可以免除對原始碼的擔保責任。

除了可享用的權利外，GPLv2 授權條款也限定被授權人的義務，當被授權人散布程式，需要將相同的權利移轉給其他被授權人，並將 GPLv2 授權條款的規定告知後續的被授權人；此外，GPLv2 授權條款的前言提到，如果自由軟體的被授權人以個人名義獲得專利時，相當於將軟體變為私有，這種情形並非社群所樂見，因此聲明任何軟體專利必須以允許每個人自由使用為前提，否則就不准許有專利。

⁴⁵ GNU, *GNU General Public License, version 2*, <http://www.gnu.org/licenses/old-licenses/gpl-2.0.html> (last visited May 10, 2010) .

⁴⁶ CLDP -Linux 中文文件計劃: <http://www.linux.org.tw/CLDP/OLD/doc/GPL.html> (最後點閱: 2010/5/10)。

⁴⁷ Kuo-lien Hsieh, 「How Does the GNU General Public License Demonstrate Idealism in the Free Software Community?」, 世新法學, 第 1 卷第 1 期, 頁 1-199-1-220(2007)。

5.2.2 GPLv2 第 5 條⁴⁸

5.2.2.1 條款規定

您沒有在簽署此份授權條款，因而您不一定要接受此授權條款。但是沒有其他東西得賦予您對程式與其衍生著作進行改作與散布的權利。如果您不接受此授權條款，這些行為便是法律所禁止的。因此，**如果您對程式碼或程式碼的衍生著作進行改作或散布，就代表您接受這授權條款，以及它所有關於對程式碼或衍生著作進行複製、散布和改作的條款和條件。**

5.2.2.2 廠商使用 Embedded Linux 的情況

嵌入式產品的開發廠商利用 Linux 作業系統的原始碼來開發系統程式，在開發過程中對 Linux 作業系統的原始碼加以改作、複製與散布，因此符合了 GPLv2 授權條款第五條所規範的條件，所以嵌入式產品廠商在使用 Linux 作業系統的原始碼時，雖然並未簽署同意 GPLv2 授權條款的規範，但其利用 Embedded Linux 的行為，將被視為同意受到 GPLv2 授權條款所規範。但廠商實際上是否真的注意到此項條款規定，即，即便廠商並未簽署任何開放原始碼授權條款，但廠商自身的行為便可成立授權條款。抑或廠商認為，自己並沒有實際上簽署開放原始碼授權條款，並因此而認為授權契約未成立。因此，我國嵌入式產品開發廠商在實務上，對於 GPLv2 授權條款的實際認知情形仍有待瞭解。

5.3 GPLv2 授權條款中與嵌入式產品特質相關的條文

以下分別說明嵌入式產品的本質中，特別容易違反 GPLv2 條款規定的條款，架構上首先簡要說明各個條款的規範內容，其次說明現行的嵌入式產品開發廠商的作法，並比較廠商作法與 GPLv2 授權條款的差異，詳細的條款內容則請參看附錄三。

⁴⁸ GPLv2 §5 You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

5.3.1 GPLv2 第 0 條⁴⁹

5.3.1.1 條款規定：

凡是**著作權的所有權人**在程式碼和作品中聲明其程式碼和作品是以 GNU GPL 授權條款作為授權依據而散布，這些程式碼或作品都受到本授權條款的約束。…本授權條款不適用於**重製，散布和改作**以外的行為，這些行為超出這些條款的範圍，執行本程式的行為不受條款的限制。…

這個條款提到，著作權人在散布其程式或作品時，可以選擇以 GPLv2 授權條款作為授權依據，而被授權人使用以 GPLv2 授權條款的程式（原始碼、目的碼）時，非重製、散布和改作的行為並不受到限制。換句話說，GPLv2 授權條款係以著作權法賦予著作權人的重製、散布與改作的權利為限制對象，而後續在 GPLv2 授權條款中的第 1~3 條中，則再進一步的針對這三種情形作進一步的說明。

5.3.1.2 廠商使用 Embedded Linux 的情況

由於嵌入式產品開發廠商使用 Embedded Linux 的目的，便是將 Linux 作業系統的原始碼，應用於自己產品的系統程式，並搭配相關的應用程式及硬體驅動程式使用。因此無論是晶片設計公司或系統開發廠商，嵌入式產品開發廠商在開發系統程式時，因為會對 Linux 作業系統的程式進行修改，便符合此條款中關於改作的規定；再者，嵌入式產品開發廠商在販售控制晶片，或嵌入式產品的成品時，便符合散布的要件，同樣的，因為在散布程式時，目的碼必須被燒錄至非揮發性記憶體中，因此關於重製的條件也同樣成立。

換句話說，若是依照標準的 GPLv2 授權條款第 0 條的規範，廠商在生產並販售嵌入式產品時，便符合”散布程式”的定義，因為系統程式的目的碼，係被儲存於非揮發性記憶裝置內，即便一般的使用者不直接看到系統程式的原始碼，甚至不會意識到自己的操作流程需要透過軟體來實現，但此種類型的散布方式，仍然落入 GPLv2 授權條款所規定的類型，因此廠商亦不得限制程式的接收者，即購買嵌入式產品的消費者的權利。

⁴⁹ GPLv2 §0: This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. ...Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted...

就個人電腦的使用而言，程式的散布往往代表軟體的複製，無論是本機操作時的複製（複製到隨身碟），或是透過網路（下載、上傳、e-mail 附件夾檔）等，且散布之後的執行平台往往不需要特別考量便可以使用被散布的軟體，因此，在個人電腦的平台上，程式被散布時的可再被利用的情形也較為普遍。

就嵌入式產品而言，一般的使用者（消費者）無法對嵌入式產品的程式進行複製與散布，因為程式被燒錄於非揮發性裝置中，而其讀取的方式並不像個人電腦容易使用，因此一般使用者並沒有能力複製程式，更無法散布程式的原始碼，只能執行產品上既有的程式。亦即，就嵌入式產品的應用而言，嵌入式產品的開發廠商才是真正有能力對原始碼進行重製、散布、改作程式的主體，也是 GPLv2 授權條款所欲規範的對象。

5.3.2 GPLv2 第 1 條⁵⁰

5.3.2.1 條款規定

1. 只要您在每一程式副本上明顯和恰當地**宣告著作權聲明**和**免除擔保的聲明**，並**原封不動保持此授權條款的聲明和免除擔保的聲明**，並**將此授權條款連同程式碼一起給其他每位程式接受者**，您就可以利用任何媒體複製，和發布您收到的程式的原始碼。

您可以為轉讓副本的實際行動收取一定費用。您也可以自由決定是否以提供擔保來換取一定的費用。

GPLv2 授權條款的第一條主要規定被授權人在複製與發布程式時，必須宣告著作權聲明與免除擔保的聲明，並提供 GPLv2 授權條款與原始碼給程式的接收者。以嵌入式產品而言，廠商所散布的程式通常為燒錄在非揮發式記憶裝置的目的碼，當被授權人在散布程式時，應該要傳遞給後手的內容應該包含：著作權聲明、免除擔保的聲明、GPLv2 授權條款的聲明以及程式的原始碼。

5.3.2.2 廠商使用 Embedded Linux 的情況

廠商在使用商業授權軟體進行嵌入式產品開發時，即便支付了軟體的授權費用，但

⁵⁰ GPLv2 §1: You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program. You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

軟體的使用範圍仍可能受到限制。舉例來說，同一個被授權廠商若有硬體設計相當類似的產品，但分屬不同型號時，若軟體採用的是商業授權條款時，被授權廠商可能需要重新對第二種產品洽談軟體授權，並另外支付一筆軟體授權費用。因此，即便廠商對硬體功能的修改微乎其微，卻必須額外支付一筆龐大數額的授權金，加上軟體本身並不是廠商用來販售的標的，因此廠商當然希望開發成本能越低越好。在銷售嵌入式產品時，除了以嵌入式產品開發廠商作為服務對象的開放原始碼軟體的軟體開發廠商，得以透過對軟體支援所提供的服務而獲利外，一般銷售嵌入式產品的廠商並不能利用軟體來賺取利潤，因此嵌入式產品的開發廠商自然會選用不需要授權金的開放原始碼。

當廠商在使用商業授權軟體時，對外提供程式時，便受到廠商與軟體授權者間的授權條款所規範。同理，當廠商使用的是開放原始碼時，自然應該受到開放原始碼授權條款的規範，但是 GPLv2 授權條款的第一條規定廠商在散布程式時，即晶片設計公司銷售晶片給系統開發廠商時，必須提供範例程式而產生散布的行為，以及當系統開發廠商在販售嵌入式產品所構成的散布，都必需要遵守規定而提供著作權聲明、免除擔保的聲明、GPLv2 授權條款的聲明，以及程式的原始碼給一般的使用者，此種規定對於嵌入式產品開發廠商而言，非但不同於傳統的授權條款，更帶來非預期的影響。

對嵌入式產品廠商來說，提供系統程式的原始碼是一件非同小可的事情，一般來說，會關心原始碼的消費者相當少見，遑論是看得懂原始碼的消費者，即便看得懂原始碼的語法，但系統程式的原始碼相當繁複，在對嵌入式產品的硬體架構不了解，或是對系統控制流程不熟悉的情況下，即便消費者具有程式撰寫經驗，但系統程式的了解也非一蹴可幾的。事實上，即使是參與嵌入式產品開發的系統工程師，在現行開發模式下，由於採用分層與分模組的分工作法，各種功能模組的負責人，固然瞭解自己負責之功能模組的運作方式，但對於他人負責的原始碼也經常是一問三不知，更何況是一般的消費者，因此有些廠商認為，即便提供原始碼給消費者，消費者不懂，也沒有能力去理解程式，因此提供原始碼是不必要的。

因此，廠商會認定，一旦自己依照 GPLv2 授權條款的規定，提供系統程式的原始碼時，最可能因此而獲得利益的並非一般的消費者，而是類似產品的競爭對手。即便各家廠商的硬體架構不同，系統程式的原始碼也無法直接套用，但是概念與構想，如：特殊演算法的使用，卻可以透過原始碼的提供而外洩。實務上也可以發現，當前的嵌入式產品廠商在使用開放原始碼時，鮮少會主動提供原始碼，也使開放原始碼授權條款的效力並未完全發揮而喪失讓程式交流的初衷。

5.3.3 GPLv2 第 2 條⁵¹

5.3.3.1 條款規定

2. 您可以修改程式的一個或幾個副本或程式的任何部分，以此形成基於這些程式的衍生著作。只要您同時滿足下面的所有條件，您就可以依照第一款的要求複製和散布這一經過修改的程式或著作。

a) 您必須在修改過的檔案中附有明顯的說明：您修改了此一檔案及任何修改的日期。

b) 您必須讓您發佈或出版的著作，**包括本程式的全部或一部分，或內含本程式的全部或部分所衍生的著作，允許第三方在此授權條款下使用**，並且不得因為此項授權行為而收費。

c) ...⁵² (略)

這些要求適用於**整份修改過的著作**。如果能夠確定著作的一部分並非本程式的衍生著作，且可以合理地單獨考慮並將它與原本的著作分開的話，則當您將它**以獨立的著作散布時，它不受此授權條款的約束**。但是當您將這部分的著作與基於本程式的著作一同發佈時，則**整個套件將受到本授權條款的約束**，因為本授權條款對於其他授權條款持有人的授權擴大到整個產品，也就是套件的每部分，無論其作者為何。

⁵¹ You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License. ...

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

⁵² §2(c)主要規範以互動操作介面呈現程式碼時的聲明顯示方式。

因此，本條款的意圖不在於剝奪您對完全由您自身完成著作的權利，而是透過權利的行使，來控制基於本程式的集體著作或衍生著作的散布。

此外，將與本程式無關的著作，和本程式(或本程式的衍生著作)一起放在同樣的貯存媒體或發佈媒體上，並不導致該些著作受到本授權條款的約束。

簡單歸納此條款的規定，可以發現 GPLv2 授權條款除了要求被授權人在散布程式（此處的程式係泛指原始碼與目的碼）時，必須讓第三方可根據相同授權條款使用程式。再者，被授權人在散布程式時，不得收取程式的授權費用，對於自己所修改的程式內容，也必須將程式中被修改的部份加以標示，讓後續的被授權人得以清楚辨別程式的來源，也就是程式在散布過程中，程式中的哪個部份經過何人、何時被修改過，均需要讓下游的被授權人知悉。此外，本條款認為，GPLv2 授權條款的使用，是爲了確保基於 GPLv2 授權條款而開發的程式，可以藉由權利的行使來控制散布衍生著作時，仍需遵守授權條款的規定。

需注意的是，儘管在散布程式時，「程式」一詞可能代表原始碼或目的碼，但是提到應提供「程式」讓第三方可以使用時，此處的「程式」係指原始碼，這是因爲一般人無法透過閱讀目的碼而理解程式，也無法直接修改目的碼，因此所謂的在程式內容裡頭標示、提供程式給後續被授權人時，程式一詞則專指原始碼。

GPLv2 授權條款除了對衍生著作發生效力，對於彼此相關的程式，即使不屬於衍生著作，也可能產生授權效力發生感染的效果。如果程式中的一部分與他部份相關，但此部份程式並非以 GPLv2 授權條款之創作而衍生的著作，在可被單獨考慮且可清楚與整份衍生著作區隔的情況，若該份可獨立存在之程式並未被區隔，並與整份衍生著作一併散布時，則該份程式將一併受到 GPLv2 授權條款的規範；但若該份程式與整份衍生著作分開散布時，該份程式便可不落入 GPLv2 授權條款的規範。

換句話說，若是 a 程式與 b 程式相關，而 b 程式係以 GPLv2 進行授權，但 a 程式並非由 b 程式修改而得，則 a 程式的散布方式，將影響 a 程式是否受到 GPLv2 授權條款規範之認定。假設 a 程式與 b 程式一併被散布時，a 程式便受到 GPLv2 授權條款的限制；另一方面，若 a 程式與 b 程式被分開散布時，則 a 程式便不會受到 GPLv2 授權條款的規範。

舉例來說，如果嵌入式產品中，提供了 b 程式（目的碼），之後被授權人再另外以手動方式自網路連線中取得 a 程式（目的碼）時，a 程式與 b 程式便屬於分開散布的態樣，此時 b 程式雖然採用了 GPLv2 授權條款，但其著佐權的效力並不會影響、擴及至 a 程式。

5.3.3.2 廠商使用 Embedded Linux 的情況

依照 GPLv2 授權條款的嚴格解釋，廠商一旦使用 Embedded Linux，則廠商開發出來的系統程式，無論是廠商自行開發的應用程式、針對硬體控制所撰寫的驅動程式，或是對 Embedded Linux 的原始碼所進行的修改，都必需要提供給後續的被授權人，也就是購買嵌入式產品的消費者，並提供相關的資料⁵³。也就是說，嵌入式產品的開發廠商應該讓消費者可以查看系統程式的原始碼，並在所提供的原始碼中，清楚的區分各個檔案的來源。但是在實務上，終端的消費者只會看到最終的產品，在該些產品中，包含了存有目的碼之記憶裝置，亦即，即便消費者將嵌入式產品拆卸開來，也僅會看到非揮發性記憶裝置的實體外觀，而無法讀取程式的內容，此外，若要對程式的目的碼進行讀取時，尚需透過特殊的配備方得讀取，遑論編譯前之原始碼。

即使是具有合作關係的晶片設計公司與系統開發廠商間，晶片設計商在提供程式給客戶時，也會盡量以目的碼的方式提供，因此系統開發廠商對整體系統程式的認識也相當有限。換言之，廠商對於提供原始碼給彼此簽有保密協定(Non-disclosure Agreement，簡稱為 NDA)的商業夥伴都如此小心謹慎，更何況要提供原始碼給單純購買終端產品的一般消費者。此外，廠商在販售產品給客戶時並未使用 GPLv2 授權條款，而使用另外簽訂的授權內容。

廠商使用開放原始碼的目的是希望自己的產品開發能夠借助他人的力量，也就是站在巨人的肩膀上來幫助自己，但廠商卻不希望成為幫助他人的助力，認為依照 GPLv2 授權條款的規定而提供原始碼時，該些原始碼將被競爭對手所利用，因此相當排斥讓自己投注的心力成為他人足下的肩膀。換言之，我國廠商並不希望因為使用了以 GPLv2 授權條款進行授權的 Linux 作業系統後，就受到著佐權的影響，更對原始碼受到感染一事感到相當畏懼。

就系統程式的劃分來說，作業系統屬於核心的程式，因此，作業系統無法獨立於下層的驅動程式及上層的應用程式之外，換句話說，一旦系統程式缺少了作業系統，系統程式無法順利進行操作，因此分開散布的做法對於作業系統而言並不實際，這也導致使用了 Embedded Linux 的嵌入式產品，其系統程式必然受到 GPLv2 授權條款的規範，而無法以獨立方式散布而避開著佐權效力的影響。

⁵³ Bradley C. Wright, *Linux Users Risk Infringement*, 13(8) Intellectual Property & Technology Law Journal, 1,7 (2001).

5.3.4 GPLv2 第 3 條⁵⁴

5.3.4.1 條款規定

您可以以目的碼或可執行形式複製或發佈程式（或符合第 2 款，本程式的衍生作品），只要您遵守前條第 1、2 款，並同時滿足下列三條中的任一條。

a) 在通常用作軟體交換的媒體上，**和目的碼一起附有機器可讀的完整的本程式原始碼**。…(略)

b) 在通常用作軟體交換的媒體上，和目標碼一起，**附上書面報價，提供替第三方複製原始碼的服務**。…(略)

c) …(略)⁵⁵

作品的原始碼指的是對作品進行修改最優先擇取的形式。對可執行的作品而言，完整的原始碼套件包括：**所有模組的所有原始程式，加上有關的介面的定義，加上控制可執行作品的安裝和編譯的設定檔**。至於那些通常伴隨著執行本程式所需的作業系統元件（如編譯器、核心等）而發佈的軟體（不論是原始碼或可執行碼），則不在本授權條款要求以程式原始碼形式伴隨發佈之列，除非它是本程式的一部分。 …

廠商提供原始碼的方式可採用下列三種方式：將程式的原始碼與目的碼同時提供給被授權人、提供複製原始碼所需的報價，且報價的有效期間必須不短於三年，或提供自己所收到的報價(第三種方式僅適用於非商業用途)。換言之，GPLv2 授權條款並不要求

⁵⁴ GPLv2 §3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c)...

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

⁵⁵ 只適用於非商業性發佈，在此略過。

廠商必須全面公開原始碼，而是要求廠商必須提供取得原始碼的管道給被授權人。此外，廠商除了提供原始碼的相關模組與使用介面等資訊外，還需要提供原始碼在編譯與安裝執行時的環境設定檔案，藉以輔助被授權人實際使用原始碼時使用。

至於編譯器，由於現在很多開發環境都是使用 GNU 的編譯器為基礎，因此 GNU 也鼓勵被授權人在提供原始碼給後手時，也一併提供相關的輔助資訊，以便利後續被授權人對原始碼的使用，讓原始碼的分享產生最大的效益。即便被授權人使用的是商用版本的編譯器，例如微軟的 Visual Studio，被授權人固然無權將編譯器也一併提供給後端之被授權人，但開放原始碼的社群仍希望被授權人可以提供相關的操作說明給後續的被授權人。

5.3.4.2 廠商使用 Embedded Linux 的情況

在我國的實務中，嵌入式產品的開發廠商即便使用的 Linux 作業系統，也未提供系統程式的原始碼，亦未提供原始碼的取得管道，即便是在晶片設計公司與系統開發廠商間，晶片設計公司在提供程式時，往往以目的碼搭配原始碼的方式來提供。亦即，晶片設計公司所提供的原始碼，經常限於系統開發廠商要修改的功能或應用，一旦涉及晶片設計公司希望保留的功能，或是由第三方授權而來的程式時，則晶片設計廠商將以目的碼搭配標頭檔的方式提供給系統開發廠商，而不會提供原始碼。

就開發環境來說，系統開發廠商會對原始碼進行修改，並進行功能的驗證，因此晶片廠商必須將工具程式（編譯器、binutils、C 函式庫）以軟體開發套件（SDK）的方式提供給系統開發廠商，但是系統開發廠商在提供產品給品牌廠商或是產品使用者時，便不會提供相關的環境。

5.3.5 GPLv2 第 4 條⁵⁶

5.3.5.1 條款規定

4. 除了本授權條款明白聲明的方式之外，您不能複製，修改，再授權和散布程式。任何嘗試以其他方式重製，改作，再授權和散布程式的行為是無效的，而且將自動結束

⁵⁶ GPLv2 §4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

授權條款賦予您的權利。...

根據此條款的規定，嵌入式產品的開發廠商在不遵守 GPLv2 授權條款的情況下對於相關程式的進行複製、改作、再授權與散布，都將讓嵌入式產品的開發廠商自動喪失使用程式的權利。

也就是說，儘管嵌入式產品的開發廠商可以自由的取得以 GPLv2 授權條款進行授權的開放原始碼，並對其加以改作，但是嵌入式產品的開發廠商一旦未遵循 GPLv2 授權條款的規定後，若接續對採用 GPLv2 授權條款的程式進行複製、改作、再授權與散布時，依據此條款，該些後續的使用也被認為違反 GPLv2 授權條款的規定。

5.3.5.2 廠商使用 Embedded Linux 的情況

我國嵌入式產品開發廠商並未注意這方面的限制，不但未遵守關於複製，修改，再授權和散布程式的規定，對於因為未遵守規定所造成不得使用原始碼的規定也不清楚。事實上，根據這個條款的規定，倘若嵌入式產品的開發廠商並未遵守 GPLv2 授權條款關於複製、修改、再授權、散布的規定，便代表嵌入式產品的開發廠商違反 GPLv2 授權條款，因此嵌入式產品的開發廠商後續所做的任何行為都被視為是違反 GPLv2 授權條款的。

5.3.6 GPLv2 第 6 條⁵⁷

5.3.6.1 條款規定

6. 每當您重新發佈程式（或任何程式的衍生著作）時，接受者自動從原始授權條款授權人那裡接到受這些條款和條件支配的對本程式進行複製、散布或改作的許可。您不可以增加任何條款來進一步限制本授權條款賦予他們的權利。...

廠商在散布以 GPLv2 作為授權條款的程式或其衍生著作時，不得對被授權人額外增添條件，或對 GPLv2 授權條款的內容進行修改、增加限制。基於這個條款的規定，當被授權人除了以 GPLv2 授權條款的軟體外，亦使用了不同授權條款的軟體時，若其他的授權條款採用的限制對於 GPLv2 授權條款不相容時（例如：商用授權條款），被授權人便

⁵⁷ GPLv2 §6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

不得以其他種類之授權條款作為 GPLv2 授權條款的限制，此時該授權條款便與 GPLv2 授權條款發生衝突。目前關於其他開放原始碼授權條款與 GPLv2 授權條款之間的衝突問題，通常會是由他種授權條款加以修改並提供雙重授權的方式來處理⁵⁸。也因此，GPLv2 授權條款成爲一個他種授權條款必須配合而加以修改的條款。

目前關於同時使用以 GPLv2 授權條款以及以其他種類的授權條款的原始碼的情況下，除了使用 BSD 三條款的原始碼不會產生衝突外，若是搭配使用的原始碼採用其他授權條款時，對於該些原始碼的授權條款內容都必須加以檢視並判斷相容性，不能直接將同一套軟體中的部份原始碼使用 GPLv2 授權條款，部份原始碼使用其他種類的授權條款，無論其他部份的原始碼是透過商用授權條款，或是其他種類的開放原始碼授權條款進行授權的皆然，否則就會違反 GPLv2 授權條款第六條的規定，形成對 GPLv2 授權條款的限制。

5.3.3.1 廠商使用 Embedded Linux 的情況

簡單來說，依照現行的發展模式，公司內部接觸到開放原始碼的人員與實際上在處理與客戶或供應商之間授權的人員並不相同。事實上，法務部門會對軟體授權條款進行瞭解的情形，往往是在研發部門需要購入商用授權軟體的時候，至於採用開放原始碼授權條款時，則可能因爲不需要支付授權費用而不會設有簽核流程，因此開放原始碼的取得可能只是由研發部門透過網路搜尋而取得，導致法務部門不清楚公司的產品在研發時，可能使用到的開放原始碼有哪些。

也因此，廠商在對外散布軟體(販售產品)時，往往以簽訂的商業合約作爲規範，根據合約而決定原始碼的提供與否，以及軟體發生問題時的責任歸屬，但廠商在討論彼此間的合作關係以及簽約時，涉入授權相關討論的往往是專案管理者(Project Manager，簡稱爲 PM)或法務部門，對於在系統程式中，是否使用開放原始碼，以及一旦使用時，相對應的開放原始碼授權條款的規定並不清楚，自然也不會在商業條款中，特別注意到開放原始碼授權條款的規定是否被確實的遵守。

5.3.7 小結

根據前述的說明可以發現，除了 GPLv2 授權條款第 0 條係規定規範的行爲外，在 GPLv2 授權條款中，與嵌入式產品特質相關而使廠商容易違反的條款主要與 GPLv2 授權

⁵⁸ MIKKO VALIMAKI, THE RISE OF OPEN SOURCE LICENSING – A CHALLENGE TO THE USE OF INTELLECTUAL PROPERTY IN THE SOFTWARE INDUSTRY 140-141, 2005.

條款規定中的著佐權規定相關，即，第 1~4 條、第 6 條。在 GPLv2 授權條款的第 1~3 條規範中，要求程式的原始碼，無論是未經修改的原始著作或修改後的衍生著作，都需要採用 GPLv2 授權條款。而 GPLv2 授權條款的第 4 條則規範，當被授權人違反了 GPLv2 授權條款的規定時，其授權便自然終止。至於 GPLv2 授權條款的第 6 條則規定當程式被散布給後手時，後手直接取得 GPLv2 授權條款所賦予之權利，並不會因為原本的被授權人未提供相關資訊給後手，而使後手被排除於 GPLv2 授權條款的效力外，因此，原始的被授權人在散布程式時，GPLv2 授權條款的效力也自動的發生轉移。

5.4 非與嵌入式產品特性相關的條款

前述的幾個 GPLv2 條款是目前嵌入式產品開發廠商最容易違反的規定，至於其他的條款，雖然嵌入式產品也可能發生違反的情形，但是基本的考量與其他非嵌入式產品的應用相類似，因此就不特別一一進行說明，僅說明主要規範的意含。

在 GPLv2 授權條款的第 7 條與第 8 條中，說明了關於專利或其他可能衍生的法律規範，導致 GPLv2 授權條款與法律的衝突等情形，其規範意旨主要為，當被授權人因為 GPLv2 授權條款的規定與法律的規定有所衝突時，就應該放棄使用 GPLv2 授權條款的原始碼。GPLv2 授權條款的第 9 條與第 10 條則分別對 GNU 可能發行的 GPL 後續版本，針對未來較新版的 GPL 授權條款，與 GPLv2 授權條款間的適用情形加以說明，以及 GPLv2 授權條款與其他開放原始碼授權條款的相容性，應該由原始碼的著作權人來認定，並非由被授權人自行判斷。

GPLv2 授權條款的第 11 條與第 12 條的條款與原始碼提供者的免除擔保責任相關，其中第 11 條規定因為原始碼的提供者採用免費授權的方式提供原始碼，因此在法律的許可範圍內，GPLv2 授權條款對於原始碼的功能並不負擔保責任，並由被授權人承擔使用原始碼所產生的風險，與修正時所衍生的成本。而第 12 條則規定，除非有法律或書面同意的要求，採用 GPLv2 授權條款之開放原始碼，若功能有問題而造成被授權人的損失時，該原始碼之著作權人一概不需負責。因此，無論被授權人改動原始碼的動機係為產品需求或自身興趣，若被授權人自己對原始碼的改作，導致其產品無法正常運作，原始碼的提供者亦不需要為此而提供保固服務。

5.5 Linux 的自我限制或棄權

開放原始碼的開發模式相當多種⁵⁹，有的是以一個核心團體作為發展模式，有的則

⁵⁹ Audris Mokus et., *Two Case Studies of Open Source Software Development: Apache and Mozilla*, 11(3) ACM Trans. Softw. Eng. Methodol., 309,346 (2002).

是分散式的管理⁶⁰，而 Linux 作業系統的開發在最初，採用的是以原作者 Torvalds 為計畫核心的開發模式。也因此，Torvalds 對參與 Linux 作業系統開發的程式開發者 (contributors) 所造成的影響包含兩個層面⁶¹，他不但成為 Linux 作業系統的發聲者（代言人），也對社群文化模範起了強化刺激的作用，這也讓 Torvalds 在 Linux 作業系統的開放原始碼計畫中具有無可動搖的地位。

Torvalds 本人並非與 GNU 的創始者，也未參與 GPLv2 授權條款的制定。Torvalds 只是在提供 Linux 作業系統的最初版本時，因為認同 GNU 的理念而選擇以 GPLv2 授權條款來授權 Linux 作業系統的原始碼，儘管核心理念相同，但是雙方針對應該如何實現理念的作法則有不同的思維。簡單來說，Torvalds 對於程式使用自由的理念不似 GNU 般極端，因此他對 GPLv2 授權條款對於 Linux 原始碼的規範效力，與 GNU 的認定方式，常常發生不一致的情形。舉例來說，GPLv2 無遠弗屆的著佐權性質便讓 Torvalds 覺得影響層面過廣，因此 Torvalds 曾經對 Linux 作業系統的應用軟體提出棄權的宣告⁶²，藉以自我限制其權利行使的範圍。

換句話說，若對 GPLv2 授權條款採取嚴格解釋時，一旦使用以 GPLv2 授權條款進行授權的原始碼，將因為著佐權所產生的感染性而使全部的原始碼都同樣受到限制，導致原始碼亦須提供給被授權人的影響。但這樣的效果在 Linux 作業系統的應用中，卻可以利用分層架構的方式，讓上層的應用程式與使用者介面免除 GPLv2 授權條款的規範效力，也免除需要提供原始碼的威脅。當前關於這個例外的認定最著名的便是 Google 所推出的 Android 平台。

Android 平台是一套架構於 Linux 作業系統之上，並利用 Apache 授權條款的開放原始碼平台，換句話說，Android 平台本身雖然以開放原始碼授權條款提供原始碼，但 Android 平台並不要求採用該平台的被授權人必須使用相同的授權條款，因此 Android 平台上層的程式開發者，便沒有提供原始碼的義務，即，介於 Linux 作業系統與上層應用程式之間的 Android 平台，阻隔了 Linux 作業系統因為使用 GPLv2 授權條款所造成的感染效力。

⁶⁰ LUCIE GUIBAULT & OT VAN DAALLEN, OPEN SOURCE IN PRACTICE, UNRAVELLING THE MYTH AROUND OPEN SOURCE LICENSES – AN ANALYSIS FROM A DUTCH AND EUROPEAN LAW PERSPECTIVE 27-30(2006).

⁶¹ STEVEN WEBER, THE SUCCESS OF OPEN SOURCE 166-171(2004).

⁶² “The GPL requires that works "derived from" a work licensed under the GPL also be licensed under the GPL. Unfortunately what counts as a derived work can be a bit vague. As soon as you try to draw the line at derived works, the problem immediately becomes one of where do you draw the line? We ended up deciding (or maybe I ended up decreeing) that system calls would not be considered to be linking against the kernel. That is, any program running on top of Linux would not be considered covered by the GPL. This decision was made very early on and I even added a special read-me file (see Appendix B) to make sure everyone knew about it. Because of this commercial vendors can write programs for Linux without having to worry about the GPL.”, See Linus Torvalds, The Linux Edge, <http://oreilly.com/catalog/opensources/book/linus.html> (last visited Mar. 22, 2010).

需注意的是，對於 Linux 作業系統的使用，廠商固然可以透過分層的作法來隔離 Linux 作業系統的著佐權性質，讓被阻隔的程式（應用程式、驅動程式等）免除了 GPLv2 授權條款所規範的感染情形，但這樣用法的前提是，Torvalds 已經放棄了他自己的部份權利，對於其他程式的原始碼，若同樣以 GPLv2 授權條款進行授權，而原作者卻未做出類似 Torvalds 的棄權宣告時，則感染性的判定亦會不同，使用時也不一定能夠透過分層阻絕的方式而免除提供原始碼的責任。

除了對於衍生著作等認定標準不同，導致 Linux 作業系統的授權範圍與 GPLv2 授權條款的規範產生歧異外，GNU 與 Torvalds 對於開放原始碼授權條款的未來走向，以及 Linux 是否應該改採 GPLv3 授權條款等問題的意見又更加不同。舉例來說，Torvalds 與其他 Linux 核心程式的開發者，對於 GPLv3 授權條款其實是抱持相當反對的立場⁶³，甚至在一份解釋如何使用 Linux 核心程式的文件⁶⁴中，也明確宣示了 Linux 在可預見的將來將持續使用 GPLv2 授權條款而不會導入 GPLv3 授權條款。倘若被授權人選擇了以 GPLv3 作為授權條款而改寫了部份的 Linux 作業系統核心程式的原始碼，在 GPLv3 授權條款與 GPLv2 授權條款並不相容的情況下，被授權人改寫而得的衍生著作與原始的 Linux 作業系統核心程式，將產生不同版本間的 GPL 授權條款爭議。

5.6 小結

因為違反 GPLv2 授權條款的情形層出不窮，身為 FSF 法律顧問的 SFLC 也透過許多文章來宣傳應該如何遵守 GPLv2 授權條款⁶⁵，讓程式開發者用來作為使用 GPLv2 授權條款的原始碼時的參考，因此當嵌入式產品的開發廠商選擇使用 Linux 作業系統時，對於 GPLv2 授權條款的規範內容與應用情形，可以透過這些說明而有初步的瞭解。對我國廠商而言，無論使用 Embedded Linux 的動機或目的為何，一旦在嵌入式產品中使用了 Linux 作業系統時，對於 GPLv2 授權條款都有必要瞭解，才能避免因為誤用而引發後續的問題。

附帶一提的是，對於提倡開放原始碼授權條款的社群來說，金錢的支付並非他們的目的⁶⁶，他們期待的是廠商遵守開放原始碼授權條款的規定，而不期待獲取廠商的賠償。因此，當廠商對於開放原始碼授權條款的內容有疑慮時，他們會盡力以讓廠商瞭解授權條款的規定，並期待廠商能夠配合遵守。

⁶³ CNET News 網站：http://news.cnet.com/Top-Linux-programmers-pan-GPL-3/2100-7344_3-6119372.html (最後點閱時間：2010 年 3 月 22 日)。

⁶⁴ A Guide to the Kernel Development Process，這是一份在下載 Linux Kernel 的原始碼時，會一併提供的說明文件。

⁶⁵ Bradley M. Kuhn et. al., A Practical Guide to GPL Compliance, <http://www.softwarefreedom.org/resources/2008/compliance-guide.pdf> (last visited Mar. 21, 2010) .

⁶⁶ Software Freedom Law Center, A Legal Issues Primer for Open Source and Free Software Projects, *See* <http://www.softwarefreedom.org/resources/2008/foss-primer.html> (last visited Jun. 6, 2010).

以 GPLv2 授權條款來說，GNU 希望廠商能夠提供衍生著作的原始碼，因此在社群發現廠商所販售的產品違反開放原始碼授權條款的規定時，若廠商願意釐清開放原始碼授權條款的規範內容而配合，便不至於產生後續的糾紛。同樣的，當廠商在使用 Embedded Linux 時，對於 GPLv2 授權條款的規定以及對於 Torvalds 授權範圍的認知是必要的，倘若在產品開發時不清楚這些規範，且在收到來函或各種來自於社群的通知時，仍不願意瞭解授權條款的規範內容，反而採用置之不理的態度，或認為可以用金錢來打發的心態時，後續可能衍生的問題可能更嚴重。



第六章 使用開放原始碼的嵌入式產品-爭議、衝突與作法

由於開放原始碼的授權方式，打破了傳統保障著作權人的思維，因此衍生了如：經濟面、商業手段、軟體工程、法律面等，不同面向的討論。正因為開放原始碼授權條款的作法，採取了與現行法律體系格格不入的思維，導致目前衍生了各種法律爭議。

舉例來說，著作權其實賦予了被授權人合理使用的權限，但 GPLv2 授權條款卻將絕大多數之改作與引用行為，一律視為被納入開放原始碼授權條款的規範內。例如：被授權人自行開發的 A 程式包含一萬行的原始碼，而 B 程式係採用 GPLv2 授權條款，若在 A 程式的內容中，使用了大約十行的 B 程式之原始碼，依據開放原始碼支持者的看法，A 程式便被視為受到 GPLv2 授權條款的規範。這樣無限擴張的影響力，是否造成與著作權之合理使用規範的歧異，甚至形成衝突？

再者，違反開放原始碼授權條款時，究竟應該被視為違反著作權，或是被視為違反合約？此外，開放原始碼授權條款與營業秘密或競爭法是否相衝突等問題，各家看法也都尚未有定論，遑論法院的判決。

綜上所述，本研究認為，我國廠商應該更小心看待開放原始碼授權條款，因為現有的幾個判決中，可以看出德國與美國的法院對於開放原始碼授權條款的有效性，以及開放原始碼授權條款的規範內容，呈現相當支持的態度。例如：要求被告廠商支付賠償金額、配合提供原始碼、核發禁制令等，即便法院的看法尚未形成定論，但廠商仍需投注心力了解與開放原始碼授權條款相關的議題才是。

6.1 遵守 GPLv2 授權條款規定之作法

正如自由軟體的愛好者認為傳統的智慧財產權規範方式不恰當，因此在著作權保障的框架下，提出一套顛覆一般人思維的手段，廠商自然也會在 GPLv2 授權條款的限制下而發展出因應之道，目前為了避免提供系統程式的原始碼所造成之潛在損失，有些技術被廠商應用在實務，以下便對這些方式稍作說明。

6.1.1 提供程式的原始碼但用科技保護措施規避

科技保措施常是著作權人用來限制他人接觸或利用著作的手段⁶⁷，無論是美國的數位千禧年著作權法(Digital Millennium Copyright Act)、我國著作權法關於防盜拷措施的規定、歐盟的著作權法指令(European Union Copyright Directive)，以及世界智慧財產權組織著作權條約(WIPO Copyright Treaty)等，FSF 都認為這些與 DRM 相關的規定阻礙了軟體自由。

亦即，DRM所要保護的權利卻是自由軟體愛好者所希望開放的。過去在制定 GPLv2 授權條款時，並未對這個部分加以規定，導致有廠商利用科技保護措施，對 GPLv2 的授權條款內容加以規避，因此 GNU 在新版的 GPLv3 授權條款中也對此做出了反制⁶⁸。事實上，關於是否應在 GPLv3 授權條款中定義對 DRM 措施的防止規定引發相當多的爭議，因為一旦規範後，反而形成對程式使用的限制，進而抵觸四大自由⁷⁰中的自由⁷¹，而喪失全面讓軟體可以被自由使用的目的。

利用科技保護措施規避 GPLv2 授權條款的作法中，最著名的就是 Tivo 公司的作法，也因此，現在類似的作法被稱為 Tivo 化 (Tivoization)。Tivo 雖然利用以 GPLv2 作為授權條款的原始碼搭配該公司的數位影像錄影機 (digital video recorder，簡稱為 DVR)，也確實遵守 GPLv2 授權條款的規定，在該公司的網頁提供了原始碼的下載資訊，但該公司透過在軟體與硬體中設置辨識碼的方式，一旦被授權人對原始碼進行修改並編譯後，新的原始碼所產生目的碼便帶有一個新的辨識碼，新的辨識碼與硬體預設的辨識碼不一致，Tivo 的系統程式在發現預設的辨識碼與系統的辨識碼不一致時，便拒絕啟動正常的功能操作。

易言之，Tivo 的作法是一種對 GPLv2 授權條款採取陽奉陰違的手段，也就是在形式上遵守 GPLv2 授權條款的規定，配合提供原始碼，但在實際應用時，卻藉由在產品所加上的判斷機制，以防護的方式判斷被授權人是否使用改動後的原始碼，讓修改過的原始碼無法發揮效用，因此並未達到 GNU 所預期的，讓原始碼被自由使用而促進交流與進步的目的。

雖然 Tivo 的行為成為 GNU 等自由派大加撻伐的對象，但是這樣的情形對於以獲利為目標的商業公司而言，卻是一種相當本能而自然的想法，而且是一個可被預期的規避

⁶⁷ 林靜君，論科技保護措施立法對著作權公共領域之衝擊，逢甲財法所碩士論文，(2008)。

⁶⁸ Richard Stallman, *Why Upgrade to GPLv3*, <http://www.gnu.org/licenses/rms-why-gplv3.html> (last visited June 5, 2010).

⁶⁹ 參見 GPLv3 §6 的規定。

⁷⁰ The Free Software Definition, <http://www.gnu.org/philosophy/free-sw.html> (last visited May 22, 2010).

⁷¹ “The freedom to run the program, for any purpose.”

行爲。事實上，這些作法並沒有對錯，只是出於立場或個人想法不同而產生的不同聲音，一如每個人的道德標準不同，而利用法律確保大家有基本的行爲規範，只要大家的行爲確實遵守該些規範，他人也應該予以尊重。

Torvalds 亦曾表示不反對 Tivo 的行爲，他認爲自由軟體應強調互惠的概念，程式開發者願意分享自己的原始碼固然值得鼓勵，但別人是否提供原始碼則不應受到原作者的限制，因此廠商若將自由軟體應用於其他系統時，他也樂見其成。

由於 Torvalds 對於 DRM 的態度較爲友善，他也曾公開表明 Linux 作業系統的授權條款將維持不變，並不會採用對 DRM 有著進一步限制的 GPLv3 授權條款，因此目前使用 Embedded Linux 的嵌入式產品開發廠商，仍可利用 DRM 的方式來維護自身的研發機密。

6.1.2 分開散布

根據 GPLv2 授權條款第二條的規定，如果是與 GPLv2 授權條款的原始碼關聯性不強的原始碼，廠商可以使用分開散布的方式，減輕提供原始碼的責任⁷²。這個部份還可以進一步根據實作技術的不同而區分爲兩種類型：

6.1.2.1 實體上的分開散布

爲了滿足實體上的分開散布要件，廠商在開發系統程式時，應盡可能對原始碼所提供的功能加以切割，配合 GPLv2 授權條款的規定，提供被授權人系統程式中，受到著佐權影響效力而必須提供的原始碼，至於不希望提供之原始碼，則以產生目的碼的方式，另外提供給消費者。換言之，廠商可以將自行開發的原始碼予以編譯後，提供編譯後的目的碼在網頁上，讓消費者自行下載，因此結合原有的 GPLv2 授權條款的程式，以及新增功能的目的碼的過程並不是由廠商自己結合。

如此一來，依照 GPLv2 授權條款的第二條規定，廠商所自行開發功能的原始碼，就無須提供給下游的消費者，若消費者另外再進行了重製、散布等行爲，則提供原始碼的責任將轉移至消費者。也就是說，消費者必須遵守 GPLv2 授權條款的規定，而提供原始碼給自己的後手，然而，該名消費者自己並未取得新增功能之原始碼，自然無法依照 GPLv2 授權條款的規定而提供原始碼。因此，廠商若能把自己的核心程式與原本的開放原始碼區隔，並採用實體的分開散布時，便能避免自己的原始碼受到感染的情形。

⁷² 葛冬梅，GPL 的另類利用方式：「分開散布·責任轉嫁」：

http://www.openfoundry.org/component/option,com_content/Itemid,353/id,1711/lang,tw/task,view/ (最後點閱時間：2010 年 5 月 22 日)。

需要注意的是，在分開散布程式時，目的碼的下載與否必須由消費者自行決定，而不能採用自動化下載的方式，因此應用時，分開散布的對象比較適合在產品的附加功能上。對產品的主要功能而言，採用分開散布的方式時，產品可能會因為欠缺主要功能而無法正常運作，再者，由於主要功能彼此的關聯性較高，也無法因為分開散布而得以免除提供原始碼的責任。此外，這種方式的使用前提是，不同程式的原始碼可以被清楚區隔，而且下載目的碼的方式與手段不應過於繁複，否則反而可能造成消費者的反感而不願使用該產品，此為廠商需考慮的部分。

6.1.2.2 動態連結程式與靜態連結程式

簡單來說，使用函式庫的目的是為了簡化程式的操作，而將一些運算或功能透過函式提供，讓其他程式可以直接使用。當程式與函式庫之間使用靜態連結（static link）時，代表程式與靜態連結的程式庫間的相依性較高；而程式與函式庫之間使用動態連結（dynamic link）時，代表程式與動態連結的程式庫之間的相依性較低。針對相依性的不同，有些論點認為：程式彼此之間的連結方式為動態連結或是靜態連結，可能對一起散布之認定會不同，進而在判斷是否違反 GPLv2 授權條款時，也會採取不同的認定。

當被授權人所開發的程式使用了以 GPLv2 授權條款進行授權的函式庫時，倘若被授權人是利用靜態連結的方式使用函式庫，則原始碼在編譯時，函式庫也會被一併被編譯成為目的碼，這種情況下，由於函式庫與程式原始碼所編譯出的目的碼已經混合而無法區分，因此屬於 GPLv2 授權條款所規範的「部分」（portion of it）並無疑義。另一方面，若程式彼此之間係採用動態連結的方式，則因為使用 GPLv2 授權條款的函式庫程式在編譯時，並不會與使用函式庫的程式一併編譯在一起，因此函式庫的目的碼是在執行系統程式時，依據執行的狀態，在需要使用相關函式時，才被動態地載入記憶體運作。

為此，有部份人士主張，當系統程式採用動態連結方式來使用以 GPLv2 授權之函式庫時，應該將函式庫與系統程式視為分開散布⁷³。但此種見解尚未形成定論，亦未經過相關判決的檢驗，因此其解釋是否成立仍有待觀察。

6.1.2.3 延遲提供原始碼的時點

在開發嵌入式產品時，對原始碼的更動與修改本來就相當頻繁，往往在一個小錯誤

⁷³ ROD DIXON, OPEN SOURCE SOFTWARE LAW 32-34, 2004.

發生的時候，便需要對原始碼進行修正，而原始碼尚需要經過編譯、連結產生目的碼後，將重新產生的目的碼燒錄於嵌入式產品內的非揮發性記憶裝置中，該修改才能算是修改完成。倘若在數千個程式原始碼（檔案）中，對其中一個檔案的原始碼中的一行，進行小幅度的修改，便需要將全部的原始碼重新編譯，並提供更新後的原始碼版本，這對於廠商確實有執行上的難度。

實務上，廠商在提供程式給客戶時，也不會每天提供程式的更新，而係以定期更新或有重大功能修改時才提供，因為程式的更新也可能衍生其他問題，因此在經過一段時間之測試後才提供之程式也具有相對較佳之穩定性，因此程式原始碼的即時更新除了有實現的困難外，也非最好的提供手段。由於 GPLv2 授權條款並未限定提供原始碼的時點，因此廠商可以利用原始碼的更新，與提供實際提供原始碼時點之時間差，避免將最新的原始碼提供出來，讓提供原始碼所造成的負面影響得以降低。

由 Skype 的訴訟也可以發現，參與自由軟體開發的社群之所以會提出訴訟，其目的是迫使廠商配合提供原始碼，因此廠商若表達配合的舉動與意願時，社群會根據廠商的態度而決定是否繼續進行訴訟。因此就原始碼的提供時點來說，除非是因為已經被社群盯上，基本上，廠商只要在社群可以容忍的期限內，依照 GPLv2 授權條款的規定，配合提供系統程式的原始碼即可消除社群的敵對態度。

需注意的是，這種作法僅能視為是一個暫態的處理方式，以社群的想法，當廠商願意提供原始碼時，代表廠商願意配合 GPLv2 授權條款的規定，相較於完全不願意提供原始碼的廠商而言，社群也相對抱以較寬容的方式來對待，但是廠商仍應在引發社群關切前，配合提供原始碼，方能避免後續之爭訟。

6.2 違反開放原始碼授權條款所衍生的嵌入式產品訴訟

開放原始碼與一般專屬軟體只提供開發後的成品不同，開放原始碼提供了製造過程所需之材料與步驟。專屬軟體所限制的事情往往都是開放原始碼推動者所不限制甚至提倡的，例如：複製、改作與散布等，會造成違反開放原始碼授權規定的原因，往往是因為廠商在販售產品（散布程式）時未遵守各授權條款的規範，在散布遵守條款所規定應提供的事宜，基於開放原始碼授權條款的不同規定，對於程式在散布時所要提供的資訊類型也有差異，即，提供原始碼的程度，或對程式來源的標示義務等規定也有差異。以下所整理的訴訟均與嵌入式產品的應用相關，該些訴訟可以依據開放原始碼授權條款的不同而區分為：違反 GPLv2 授權條款，以及違反他種開放原始碼授權條款兩種類型。

6.2.1 違反 GPLv2 所衍生的訴訟

開放原始碼授權條款藉由契約條款的規定，在原有的著作權體制下設計一套讓被授權人配合提供原始碼的措施，但是在自由使用的本質下，也連帶衍生執行開放原始碼授權條款時的問題，特別是在被授權人不特定的情況下，如何能確保每個被授權人都確實遵守開放原始碼授權條款的規定，並讓開放原始碼授權條款發揮真正的效益，都突顯了將開放原始碼授權條款應用在真實世界時，所遭遇的困境⁷⁴。

根據 GPLv2 的第二條(b)規定，一旦被授權人在散布採用 GPLv2 授權條款的軟體時，一定要使用同樣的 GPLv2 授權條款，且根據 GPLv2 授權條款第三條的規定，廠商在散布程式的目的碼或可執行檔時，必須將系統程式的原始碼（或取得原始碼的方式）一併提供給被授權人。一般來說，廠商會違反 GPLv2 授權條款情形的態樣，主要是因為廠商違反了 GPLv2 授權條款第二條與第三條的規定，即，廠商並未依照 GPLv2 授權條款的規定，將原始碼提供給購買嵌入式產品的消費者。連帶的，依照 GPLv2 授權條款第四條的規定，廠商將喪失使用以 GPLv2 授權條款授權的軟體的權利。

儘管與開放原始碼授權條款相關的糾紛不少，但在廠商自知理虧或是不願意繼續纏訟的情況下，大多會在訴訟發生後，選擇配合遵守開放原始碼授權條款的規定，而以和解收場，尚未能由法院的看法得出定論。目前與嵌入式產品相關的開放原始碼糾紛中，少數的幾個判決係以德國與美國為主。

以下提及的幾個與開放碼授權條款相關訴訟包含：Welte 在德國所提出的幾件訴訟，以及 SFLC 在美國之訴訟。Welte 是一位相當激進的自由軟體開發者⁷⁵，亦為舉報違反 GPL 授權條款情形之交流網站⁷⁷的負責人；而 SFLC 則提供 FSF 相關的法律諮詢服務，SFLC 初期的手段較為和緩，但近年來較常採用訴訟手段，目前 SFLC 所提出的訴訟大多與 Busybox 相關。

6.2.1.1 Sitcom

本案由 Welte 在 2004 年提起。Netfilter/iptables 為 Linux 作業系統的防火牆程式，此

⁷⁴ 例如，在 A 國的原始作者利用 a 授權條款將原始碼提供於網路上，該原始碼經過在 B 國、C 國等地的作者不斷地修改，後來 D 國的作者在使用該份原始碼時，違反了 a 授權條款的規定，此時應該由何人、在何地主張權利等問題相當難以確認，也不易落實開放原始碼授權條款的規定，參見：王凱，「開放原始碼授權模式分析」，產業透析：軟體與應用透析，第 12 期，頁 2-10（2003）。

⁷⁵ Harald Welte's blog, <http://laforge.gnumonks.org/weblog/index.html> (last visited Dec. 27, 2009) .

⁷⁶ WIKIPEDIA, Harald Welte, http://en.wikipedia.org/wiki/Harald_Welte (last visited Dec. 27, 2009).

⁷⁷ GPL violation, <http://gpl-violations.org/> (last visited Apr. 13, 2010).

程式採用 GPLv2 作為授權條款，但廠商們在使用時卻未遵守 GPLv2 的授權條款規定，屬於 Netfilter/Iptables 計畫成員的 Welte，便以著作權人的身份對該些廠商發出警告信。大多數的廠商因為自知理虧，而選擇以和解的方式來處理。但 Sitecom 卻對 Welte 所提出的警告信置之不理，因此 Welte 便向慕尼黑地方法院提起假處分的聲請。

慕尼黑地方法院核可了假處分的裁定，要求 Sitecom 遵守 GPLv2 授權條款而標示關於程式的取得方式與該程式係以 GPLv2 授權條款進行授權。儘管 Sitecom 提出了抗告，但慕尼黑地方法院在最終的裁定理由書中仍認為 Sitecom 不但侵犯了 Welte 的著作權，也違反了 GPLv2 授權條款，並說明 Sitecom 必須遵守 GPLv2 授權條款才能提供系爭軟體的下載，否則便構成侵權。

6.2.1.2 Fortinet UK Ltd.⁷⁸

本案由 Welte 在 2005 年向慕尼黑地方法院提出告訴，其主張為，Fortinet Ltd. 使用了 GPLv2 授權條款的軟體於其產品中，並利用加密的手段來隱匿該公司使用開放原始碼的情形。以 Welte 為首的 GPL-Violations.org 便提出告訴，主張被告公司的 FortiOS 作業系統使用了 Linux 作業系統的核心程式以及許多其他採用 GPLv2 授權的程式。

針對本案，慕尼黑地方法院對 Fortinet 核發了臨時禁制令(preliminary injunction)，要求該公司在確切遵守 GPLv2 授權條款的規定前，不得再販售（散布）包含該等軟體的產品，在此壓力下，最後 Fortinet 同意遵守 GPLv2 授權條款的規定，該公司同意在其販售的產品中，標示該些產品使用了以 GPLv2 授權條款進行授權的軟體，並同意若有消費者提出取得原始碼的需求時，僅收取複製與散布的工本費，因此兩造達成和解⁷⁹。

6.2.1.3 D-link

本案同樣由 Welte 所提出，本案涉及了 msdosfs/initrd 與 mtd 三個程式，這三個程式均屬於 Linux 作業系統的核心程式，而 D-Link 的無線網路外接儲存器(NAS)產品使用了這三個程式卻未遵守 GPLv2 授權條款的規定，而未提供相關的原始碼給消費者，且在 Welte 去函要求提供原始碼時，嚴詞否認該公司的產品使用了系爭程式。

⁷⁸ Gpl-violations.org, *gpl-violations.org project was granted a preliminary injunction against Fortinet UK Ltd.*, <http://gpl-violations.org/news/20050414-fortinet-injunction.html> (last visited May 21, 2010).

⁷⁹ Fortinet, *Fortinet (UK) Ltd. Reaches Amicable Settlement Agreement with gpl-violations.org Project*, http://www.fortinet.com/press_releases/050425_gpl.html (last visited May 21, 2010).

Welte 在使用還原工程確認 D-Link 的產品確實使用了係爭程式後，便在 2006 年向法蘭克福地方法院提起訴訟，最初 D-Link 同意停止販售係爭商品並採取修正措施，也同意支付 Welte 因為訴訟準備而產生的費用，但在之後 D-Link 卻又拒絕支付該筆費用，因此 Welte 又二度提出訴訟。

儘管第二次的訴訟的起因是 D-Link 拒絕付款，而非因為違反 GPLv2 授權條款，但在這個訴訟中，法院所作出的判決確認了 GPLv2 授權條款中所提出對於程式被授權人的散布機制符合德國法所規定的邀約與承諾，並認為 GPLv2 授權條款符合一般的契約規定，目前在 D-Link 的公司網頁也提供了相關原始碼的下載⁸⁰，似乎也可以看出在經歷過該場訴訟後，D-Link 選擇繼續遵守 GPLv2 授權條款的規定。

6.2.1.4 Skype

本案係由 Welte 在 2007 年於德國提出，Skype 在此訴訟中主張 GPLv2 授權條款違反了反壟斷法的規定，但未被法院採納，法院判決 Skype 違反 GPLv2 授權條款的規定。德國法院於本案表示，即使 Skype 認為其用戶不需要原始碼，也不可能使用這些原始碼，但該公司仍然必須遵守 GPLv2 授權規定所提出的要求而配合提供原始碼。最後 Skype 放棄上訴而同意提供 Skype 的原始碼。附帶一提的是，儘管訴訟結果已經確認 Skype 需要公開原始碼，時至今日仍未看到 Skype 為此而做出的實際行動，GNU 等團體雖然持續向 Skype 索取原始碼，但 Skype 卻僅答覆將配合提供原始碼，卻始終未提出具體的日期。

6.2.1.5 Linksys

在 Cisco 於 2003 年購併無線路由器公司 Linksys 後，Cisco 被告知其路由器的散布違反了 GPLv2 授權條款，在調查過程中也發現 Linksys 的路由器使用的是 Broadcom 的 802.11g 路由器晶片組，而 Broadcom 在稍後也承認該公司並未依照 GPLv2 授權條款的規定提供相關的原始碼。為了解決這個問題，Cisco 後來提供了 802.11 路由器軟體的開放原始碼⁸¹。由這個案件也可以看出嵌入式產品的開發過程中，即便販售嵌入式產品的廠商（即本案中的 Cisco/Linksys）並非真正違反 GPLv2 授權規定的廠商，卻可能因為供應商使用了開放原始碼，進而導致廠商間接違反 GPLv2 授權條款的現象。本案亦說明了嵌入式產品在開發時，系統程式的原始碼在各個廠商間呈現環環相扣的現象，而這個問題

⁸⁰ D-Link - GNU General Public License Disclosure, <http://tsd.dlink.com.tw/GPL.asp> (last visited May 21, 2010).

⁸¹ GPL Code Center, <http://homesupport.cisco.com/en-us/gplcodecenter>, (last visited May 22, 2010).

在個人電腦平台上，卻因為軟硬體的專業分工而不存在。

6.2.1.6 Busybox

Busybox 是一套以 GPLv2 授權條款進行授權的標準 Unix 工具，常被用在嵌入式系統的開發⁸²，目前在美國的幾起訴訟都與 Busybox 相關，該些訴訟均由 SFLC 代表 BusyBox 的開發者所提出。

1. Monsoon、Xterasys、High-Gain Antennas、Verizon Communications 等

最早因為使用 Busybox 而被提出告訴的 Monsoon Multimedia(以下簡稱爲 Monsoon)，因為在該公司的 HAVA TVplace-shifting 裝置上使用了 Busybox，卻未依照 GPLv2 的授權條款規定提供程式，因此軟體自由法律中心在 2007 年時便代表 BusyBox 的著作權人⁸³對 Monsoon 提起訴訟。Monsoon 坦承該公司並未遵守 GPLv2 的授權條款規定，並同意遵照 GPLv2 授權條款的規範，提供該公司所修改過的程式原始碼⁸⁴，因此本案以和解收場，Monsoon 除了支付和解金外，亦同意履行 GPLv2 授權條款所規範的相關義務。

除了與 Monsoon 的訴訟外，SFLC 陸續又以使用 Busybox 卻違反 GPLv2 授權條款的規定而向 Xterasys、High-Gain Antennas、Verizon Communications 等公司提出類似的訴訟，而這些被告公司在自知理虧的情況下，同樣也以和解的方式來處理。

以網路設備製造商 Xterasys 爲例，該公司同意遵守 GPLv2 授權條款的規定，提供該公司產品的原始碼、指定一名專職的員工負責對該公司的原始碼進行監控，確認與 GPLv2 授權條款的規定相符、通知先前取得該公司產品者獲取該些原始碼的途徑，並支付原告一筆數額未公開之賠償金⁸⁵。而 SFLC 與 High-Gain Antennas⁸⁶、Verizon Communications⁸⁷之間的和解條件也大致與 Xterasys 相似。

2. Best Buy 等

在 SFLC 所提起的幾件與 Busybox 相關的訴訟中，最新的一件是 SFLC 在 2009 年 12

⁸² Products/Projects Using BusyBox, <http://www.busybox.net/products.html>, (last visited May 22, 2010).

⁸³ Erik Andersen 與 Rob Landley。

⁸⁴ Linux-Watch, First U.S. GPL lawsuit heads for quick settlement, *See* <http://www.linux-watch.com/news/NS3761924232.html> (last visited Apr. 13, 2010).

⁸⁵ BusyBox Developers and Xterasys Corporation Agree to Settle GPL Lawsuit, *See* <http://www.softwarefreedom.org/news/2007/dec/17/busybox-xterasys-settlement/> (last visited Mar. 6, 2011).

⁸⁶ BusyBox Developers and High-Gain Antennas Agree to Dismiss GPL Lawsuit, *See* <http://www.softwarefreedom.org/news/2008/mar/06/busybox-hga/> (last visited Mar. 6, 2011).

⁸⁷ BusyBox Developers Agree to End GPL Lawsuit against Verizon, *See* <http://www.softwarefreedom.org/news/2008/mar/17/busybox-verizon/> (last visited Mar. 6, 2011).

月向紐約地方法院提出對十餘家廠商提出違反 GPLv2 授權條款第 2、3、4 條規定與著作權侵害的訴訟⁸⁸，值得注意的是，除了 Best Buy、SAMSUNG、JVC 等國際知名大廠外，台灣廠商合勤科技 (ZYXEL) 亦為被告之一。本案涉訟的嵌入式產品包含影音產品與網通產品，由此亦可看出，各種不同類型的嵌入式產品，都可能因為使用開放原始碼而發生爭議。

目前，在該案的被告中，Westinghouse Digital Technologies (西屋電子) 已經被法院宣告故意侵權，並被要求支付共計九萬美元的懲罰性賠償，法院也對其使用 Busybox 原始碼的高畫質電視產品，進一步核發了永久禁制令⁸⁹。目前西屋電子已經宣告破產⁹⁰，可見此訴訟對於廠商之影響。儘管本案的其他被告仍在訴訟進行過程中，但是根據法院對 GPLv2 授權條款的解讀，可以發現法院不但肯認 GPLv2 授權條款的效力，且廠商的違反行為，亦有可能被法院解讀為惡意，甚至進一步被核發禁制令與懲罰性賠償，對於廠商的影響又更大，可見相關的法律議題確實不容忽視。

在本案中涉訟的我國廠商，合勤科技，目前在該公司的網頁提供了原始碼的下載資訊⁹¹，提供其顧客取得原始碼的管道，這似乎可以代表廠商傾向息事寧人的態度，而不願意與 SFLC 就 GPLv2 授權條款之有效性與規範內容進行爭訟。當然，合勤科技選擇採用配合手段的動機不得而知，但可能推論是，根據過去 BusyBox 的相關訴訟中，SFLC 獲得勝訴，或被告的廠商選擇與 SFLC 進行和解的紀錄，自認為與 SFLC 纏訟而勝訴的機會不大，與其進行訴訟而對自家公司的產品銷售造成影響，不如避免負面影響被擴大，故而採取配合提供原始碼的手段。

6.2.2 違反他種開放原始碼授權條款所衍生的訴訟

違反 GPLv2 授權條款規定的類型，大多是廠商在進行產品的生產與銷售時，違反著作權相關條款的規定所造成的。另一方面，儘管其他的開放原始碼授權條款不一定具有 GPLv2 授權條款的強烈感染力，但也曾經有廠商因為違反其他類型的開放原始碼授權條款，而被提起訴訟，其中最著名的便是 Jacobsen 與 Katzer 的訴訟，以下僅簡單介紹該案

⁸⁸ SFLC 的新聞網頁，該訴訟的起訴文件亦可在網頁上下載，

<http://www.softwarefreedom.org/news/2009/dec/14/busybox-gpl-lawsuit/> (最後點閱時間:2010年4月11日)。

⁸⁹ Free Software Comes at a Price: Violation of Open Source License Leads to Enhanced Monetary Damages and Forfeiture of Products in Software Freedom Conservancy v. Best Buy et al., <http://www.fr.com/free-software/> (last visited Feb 7, 2011).

⁹⁰ BusyBox and the GPL Prevail Again, See <http://www.groklaw.net/article.php?story=20100803132055210> (last visited Feb 7, 2011).

⁹¹ BusyBox GPL Software Notice, <http://us.zyxel.com/Support/GPL-OSS/BusyBoxNotice.aspx> (last visited Mar 7, 2011).

之歷程，詳細之討論可以參見相關研究⁹²。

6.2.2.1 Jacobsen v. Katzer⁹³

Java Model Railroad Interface (簡稱為 JMRI) 是一個利用 Artistic 1.0 開放原始碼授權條款的軟體，該軟體被用來用來控制火車模型，Artistic1.0 授權條款的內容與解釋有其他文獻可參考⁹⁴，簡單來說，Artistic1.0 授權條款的內容規定任何被授權人皆能不受限制使用、重製、散布程式，但需遵守完整的著作權聲明與標示作者的姓名。

JMRI 計畫的主持人 Rober Jacobsen 在 2006 年 9 月向法院聲請臨時禁制令，原因是被告 Katzer 所開發的 Decoder Commander 程式利用了 JMRI 的程式並散布，卻未依循 Artistic1.0 授權條款的規定。被告亦坦承他們複製並散布了部份取自於 JMRI 的程式，並將該些程式應用於自家的程式中，且未遵守 Artistic 1.0 授權條款的規定標示出 JMRI 原始的著作權聲明。

加州北區聯邦地方法院 (U.S. District Court for the Northern District of California) 將使用原始碼的情形區分為兩種情形：分別是違反授權的範圍，以及違反授權條款的規定。若被告違反了授權條款的範圍，則被告違反著作權法；若被告違反了授權條款的規定，則被告僅是違反合約。而加州北區地方法院認為 Artistic 授權條款中，關於原作者的標示規定屬於授權條款的規定而非範圍限制，既然 Jacobsen 有意讓他人自由的使用、修改與散布其程式 (原始碼)，因此 Katzer 的行為並未違反著作權法，僅是違反了合約規定，故 Jacobson 並因著作權之侵害而遭受不可回復的損害 (irreparable harm)。Jacobsen 因為不服這樣的判決而進一步對聯邦巡迴法院提出上訴。

在 Jacobsen 所提出的上訴中，聯邦巡迴上訴法院認為 Artistic 授權條款的著作權授權契約成立，且 Jacobsen 在提供原始碼時，要求被授權人必須遵守 Artistic 授權條款，否則便不得擁有上述權利。而被告在使用 Decoder Pro 的原始碼時，並未遵守 Artistic 授權條款，即，被告使用 Decoder Pro 原始碼的方式超過 Artistic 授權條款所准許的範圍。上訴法院並認為，採用 Artistic 授權條款的方式，讓社群得以維繫原始碼的開發，因而將原判決廢棄並發回更審。在本案中，上訴法院第一次對開放原始碼授權條款進行解讀，因此本案相當具有代表性，也引起相當多的討論⁹⁵。

⁹² 陳儷卿，開放原始碼軟體授權之相關法律風險分析，交大科法所碩士論文，頁 88-105 (2010)。

⁹³ Jacobsen v. Katzer, 535 F.3d 1373, 1380-82 (Fed. Cir. 2008).

⁹⁴ ANDREW M. ST. LAURENT, UNDERSTANDING OPEN SOURCE AND FREE SOFTWARE LICENSING, 90-98 (2004).

⁹⁵ Lawrence Rosen, *Bad facts make good law: The Jacobsen case and Open Source*, 1(1) International Free and

目前本案在 2009 年 1 月由加州北區地方法院作出了終局判決⁹⁷，判決結果雖然原告獲得部份勝訴，但和原告的預期仍有不小的差距。因為加州北區地方法院認為，原告無法證明若未核發臨時禁制令的情況下，將導致原告本身不可彌補的損害，且原告係自願提供原始碼給公眾使用，當被告未依照授權條款的規定，而未實現標示義務時，原告無法證明自己將受到何種損害，法院認為這樣的情況不足以構成初步禁制令的要件，故駁回原告所提出之臨時禁制令的聲請。

6.3 使用 Embedded Linux 並遵守條款規定的作法

GPLv2 授權條款固然反對廠商藉由軟體授權的方式來收取權利金，但 GPLv2 授權條款其實不反商業行為，舉例來說，如果廠商是透過以下幾種方式來獲利，對於 GPLv2 授權條款的推廣其實更有助益，也不在禁止之列。例如：提供軟體的支援與諮詢服務、撰寫工具書、提供便於使用者操作的軟體介面，以及提供相關的測試以確保程式的效用等。易言之，GPLv2 授權條款的擁護者認為，任何廠商均可以在提供更便於開放原始碼的使用基礎上，找出相關的利基而發展出一套獲利模式，進而獲取利潤。

自由軟體在最初發展時，係以個人電腦系統或是工作站作為平台，這些平台的共通性為，軟體的市場較大，因此發展出相對應的服務或獲利模式可以在足夠數量的基礎上進行。但是嵌入式產品的先天特性是，軟體相對不若硬體受到重視，遑論利用嵌入式產品的軟體來發展出各種獲利模式。

以個人電腦的硬體裝置為例，在 IBM PC 的規格發展下，標準配備的個人電腦無論採用的 CPU、晶片組、各種介面卡或主機板廠商等，彼此之間所用的溝通協定，無論是無形的時脈定義或是有形的匯流排規格等 PCI，都有一致的標準。

也就是說，對個人電腦而言，其硬體介面係使用相同的規格，讓軟體的開發者不需要擔心軟體的適用會因為硬體開發廠商的不同而無法通用，因此 PC 產業的軟硬體分工可以相當明確，讓硬體廠商與軟體廠商都有發揮的舞台，這與嵌入式產品的開發過程中，廠商必須同時包辦硬體與軟體開發二者的情況全然不同。嵌入式產品的開發廠商既然無法銷售軟體，在開發系統程式時，更會選擇以各種手段來減少自己開發的負擔與責任，因此廠商想要規避開放原始碼授權條款的心態，其實相當容易被理解。

Open Source Software Law Review, 27, 32 (2009).

⁹⁶ Edmund J. Walsh & Andrew J. Tibbetts, *Reassessing the Benefits and Risks of Open Source Software*, Intellectual Property & Technology Law Journal, 2010.

⁹⁷ 609 F. Supp.2d 925

6.3.1 透過提供 Embedded Linux 的技術服務而獲利

由於 Linux 作業系統盛行的關係，目前有些商業公司也開始提供基於 Linux 作業系統的原始碼為基礎，而開發的作業系統相關之支援服務，而其中最著名的有 Monta Vista、Red Hat⁹⁸與 WindRiver。

與台灣嵌入式產品廠商不同的是，這些廠商是以提供軟體支援服務為主，即，台灣的嵌入式廠商所販售的商品是嵌入式產品，而 Monta Vista 等廠商則著重於開發在嵌入式產品上執行的作業系統，並以提供系統開發時的支援服務為其獲利方式。因此這些類型的廠商縱然得以自 Embedded Linux 的開發得到獲利，但其營利模式卻是針對廠商提供服務，這與我國廠商販售硬體成品給消費者的營利模式相當不同。

換句話說，台灣的嵌入式產品開發廠商是該等廠商的潛在客戶，二者的銷售對象與販售標的並不相同，因此這些以支援服務為主的公司，所採用的「分開散布」模式固然用以參酌，但基於商業模式本身的差異，相同的作法應用至嵌入式產品開發廠商時，仍會因為產品的不同與應用時的考量而未能一體適用。

以 MontaVista 為例，最初 MontaVista 採用免費提供原始碼的模式，而以服務與升級進行收費，但自 2002 年後的版本，即 MontaVista Linux 2.0 開始則必須透過付費的方式才能獲得相關的支援與升級程式、補釘 (patch)。因為 Monta Vista 選擇利用分開散布的方式而免去 GPLv2 授權條款的規範。至於同樣以商用嵌入式 Linux 平台為產品的 Wind River⁹⁹，該公司所開發的 Wind River Linux 3.0 同樣在形式上遵守 GPLv2 的規定而提供了 Linux 作業系統的原始碼，但對於相關產品的原始碼的流通與授權，則另外採取了相當嚴密的規範措施。

6.3.2 以 Linux 為基礎的各種開發平台

6.3.2.1 Openmoko 硬體平台

當前的嵌入式產品開發廠商中，對於提供系統程式的原始碼抱持最開放態度的便是 Openmoko 所提供的開發平台。甚至可以說，Openmoko 從一開始便主打此種市場¹⁰⁰。亦即，Openmoko 採取的便是取之於大眾，用之於大眾的模式，並據此而進行相關系統程式的開發，因此 Openmoko 不但將其系統程式的原始碼予以公開，甚至為了推廣

⁹⁸ See WEBER, supra note 61, at 199-201.

⁹⁹ Where Embedded Open Source Goes Business-Critical, <http://www.windriver.com/products/linux/>, (last visited May 22, 2010).

¹⁰⁰ Openmoko™ - Open. Mobile. Free, http://wiki.openmoko.org/wiki/Main_Page, (last visited May 22, 2010).

Openmoko 的平台而在大專院校設置實驗課程與開發競賽等。

儘管 Openmoko 的程式開發符合 GPLv2 授權條款的規定，也提供較其他的手機平台更多種類的作業系統與應用等，但是使用 Openmoko 手機族群也與使用其他品牌手機的消費族群相當不同，亦即，使用 Openmoko 手機的消費者大多為特定想要利用手機進行相關開發的開發者，或以實驗為目的的教育族群，而非一般的消費者。也因此，與一般流行的品牌手機利用不同系列與外型、功能而吸引顧客的手段不同的是，Openmoko 卻只推出 free Runner 與 Neo Free Runner 兩種款式作為開發平台。

由 Openmoko 的推廣手法也可以看出 Openmoko 主攻以程式開發者為主的小眾市場，即利用開放原始碼的手段來吸引程式開發者提供其心血結晶，然後出售硬體來讓程式開發者得以嘗試其開發成果，採用這種商業模式時，產品的單價較高，市場也較小，因此非主打市場佔有率的台灣廠商所好。

6.3.2.2 Android (Google)

搜尋引擎的龍頭 Google 也跨入了嵌入式產品的開發而提出了手機的開發平台，Google 本身著眼的是上層應用軟體的市場，因此將嵌入式產品視為是平台應用的基礎。也就是說，Google 藉由搭配宏達電(HTC)等系統開發廠商來推廣其 Android 開放原始碼計畫(Android Open Source Project)，讓系統開發廠商減少系統開發的成本，而 Google 也提升其上層程式的市占率。

Android 是 Google 在既有的 Linux 作業系統的基礎上開發出來的一種架構，由於底層的作業系統為 Linux，因此下層的作業系統仍須符合 GPLv2 授權條款的要求，而 Google 也無條件的提供 Android 的底層程式的原始碼。在 Linux 作業程式的上層，Google 則提供了使用 Apache 2.0 授權條款的 Android 開放原始碼計畫。換言之，系統開發廠商所使用的 Android SDK 或是自行開發的軟體並不會直接使用以 GPLv2 授權條款作為授權依據的 Linux，相當於利用 Android 開放原始碼計畫形成一個隔絕 Linux 感染效力的屏障。

Android 開放原始碼計畫的目的是提供一個共通的介面予上層的應用軟體，其程式架構與各層程式所使用的授權條款可簡單用下圖說明：

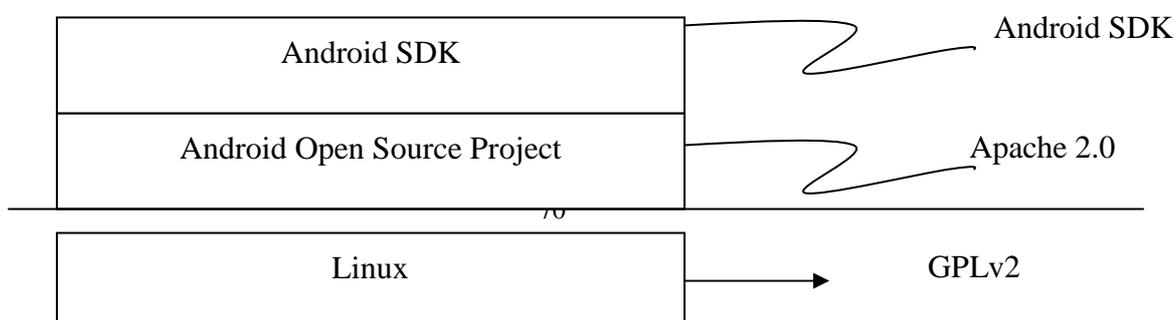


圖 4，Android 開放原始碼計畫之系統架構與所對應之開放原始碼授權條款。

資料來源：作者整理¹⁰¹

Google 所提供的 Android 不但讓嵌入式產品的開發廠商省去了開發上層應用程式的麻煩，所使用的 Linux 作業系統又是具有相當口碑、穩定度相對成熟的底層程式，因此對於講求時效的嵌入式產品市場相當具有吸引力。當廠商參與 Android 開放原始碼計畫時，不但可以使用 Linux 作業系統，利用現有的社群資源以節省開發成本外，又得以藉由 Android 開放原始碼計畫得屏障效果而免除提供原始碼的義務，因此嵌入式產品的開發廠商亦樂於接受這樣的開發模式。

屬於學術類型的 Apache 2.0 授權條款¹⁰²被視為與 GPLv2 授權條款相容¹⁰³，在 Android 開放原始碼計畫中，搭配 Android 使用的開發工具、範例程式與文件係以 Android Software Development Kit License Agreement¹⁰⁴作為授權依據，該授權條款的內容亦可見於 Android 的網站。對於參與開發 Android 開放原始碼計畫的開發者而言，根據開發者身份的不同則需要提供不同類型的 Contributor License Grants¹⁰⁵，透過這些同意書的預先簽署，也可以避免日後需要進行授權時，產生如過去因為開發者的來源不一，所導致使用開放原始碼授權條款進行授權時衍生的問題。

6.3.2.3 其他平台

除了 Android 平台外，Intel 也曾經推出使用 LGPL 授權條款的 Moblin 平台，而 Nokia 則推出 Maemo 平台，Maemo 平台並不要求被授權人使用特定的授權條款，只要使用 OSI 認可的開放原始碼授權條款即可。在 Moblin 平台與 Maemo 平台的基礎上，Intel 與 Nokia

¹⁰¹ 關於 Android 計畫選用該些開放原始碼授權條款的理由，參見 Licenses, <http://source.android.com/source/licenses.html> (最後點閱時間：2011 年 3 月 8 日)。

¹⁰² The Apache Software Foundation, Applying the Apache License, Version 2.0, <http://www.apache.org/dev/apply-license.html> (last visited Apr. 11, 2010).

¹⁰³ The Apache Software Foundation, Apache License v2.0 and GPL Compatibility, <http://www.apache.org/licenses/GPL-compatibility.html> (last visited Apr. 11, 2010).

¹⁰⁴ Android Developers, Terms and Conditions, <http://developer.android.com/sdk/terms.html> (last visited Apr. 11, 2010).

¹⁰⁵ 若開發者的身分屬於個人時，需簽署 Individual Contributor License Grant；若開發者以公司的身份參與時，則需簽署 Corporate Contributor License Grant，參見 Android Open Source Project, <http://source.android.com/license> (最後點閱時間：2010 年 4 月 11 日)。

合作開發了 Meego 平台。

Meego 平台的作法是：將其系統程式的軟體堆疊分為兩層，其中與 Linux 作業系統之核心程式（Kernel）直接相關的部分稱為 MeeGo Operating System；在 MeeGo Operating System 上層的軟體堆疊則為 MeeGo User Experience Subsystem。MeeGo 平台的特色為，軟體堆疊所使用的授權條款並未被限定。與 Linux 作業系統的核心程式直接相關的前者，可以使用所有的 OSI 相容授權條款，但鼓勵使用 GNU (L)GPL；與應用程式較為相關的後者，則必須使用某些允許性的 OSI 相容授權條款，並特別鼓勵使用具有著佐權效力的授權條款(如:GNU LGPL v2.x)作為函式庫的授權。

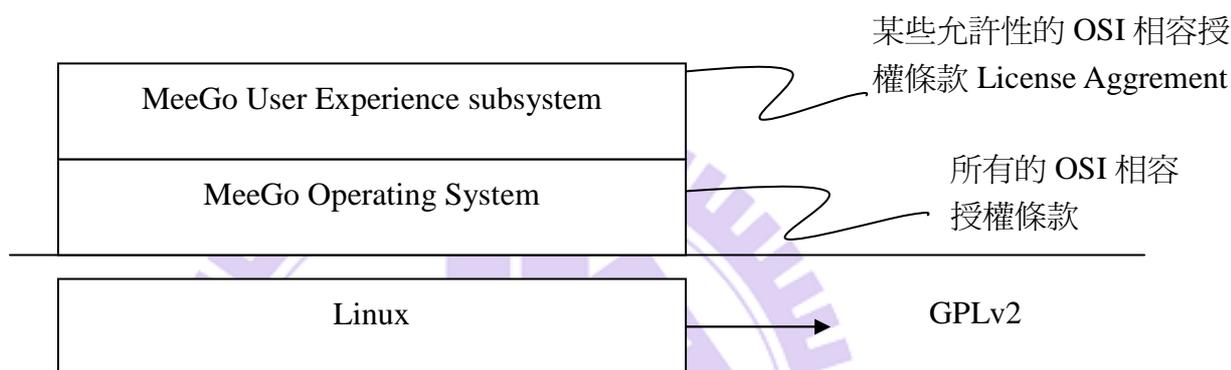


圖 5，MeeGo 開放原始碼計畫之系統架構與所對應之開放原始碼授權條款。

資料來源：作者整理¹⁰⁶

經由上述的說明可以發現，目前以 Linux 作業系統為建構基礎的作法堪稱是一種公認區隔 GPLv2 授權條款效力的作法，目前提供該些開發平台的廠商均為國外的大廠，我國國內尚未見廠商推動類似的開發平台，這也和我國廠商著重在硬體製造有極大的關係。對於這些由商業公司所提供的開發平台雖然也存在不同的意見¹⁰⁷¹⁰⁸，認為其開放程度並非真的完全提供被授權人透明的開發環境，即，不符合全面自由使用原始碼的核心理念，但其外顯的作法符合 Embedded Linux 的授權規定則無疑義。

6.4 小結

根據上述的訴訟整理可以發現，目前廠商違反開放原始碼授權條款的類型為：未依

¹⁰⁶ 關於 MeeGo 的授權策略，參見 Licensing Policy, <http://meego.com/about/licensing-policy> (最後點閱時間：2011 年 3 月 8 日)。

¹⁰⁷ Android 不是那麼開放，http://heaven.branda.to/~thinker/GinGin_CGI.py/show_id_doc/397 (最後點閱時間：2010 年 5 月 12 日)。

¹⁰⁸ Android 還是開放源碼計劃嗎?, <http://cwhuang.info/2010/01/is-android-still-an-open-source-project> (最後點閱時間：2010 年 5 月 12 日)。

照規定標示原始碼的或作者等資訊、未依照開放原始碼授權條款規定的方式提供原始碼。前者對於廠商的影響雖然較小，但廠商仍可能有其他顧慮，而不願他人知悉該產品使用了開放原始碼；而後者的使用更因為涉及釋出範圍的解讀，以及可能牽涉到不同種類授權條款之間的影响等議題而讓著重於技術開發的廠商感到困擾。這方面的議題也逐漸的受到重視¹⁰⁹。

由目前的相關爭議的處理方式，無論是和解或是有判決產生的情形，可以觀察到對於相關自由軟體授權的肯定，使得嵌入式產品廠商在違反 GPLv2 授權條款的時候，對於相關社群的要求會盡量使用拖字訣¹¹⁰來處理，基於自身理虧的立場，因此在被提出相關要求開放原始程式的聲浪固然不會拒絕，但以沒有時間表的回應方式，可以看出在這些嵌入式產品的廠商眼中，依據開放原始碼授權條款的規定而提供系統程式的原始碼是一件非同小可的事情。

開放原始碼的愛好者在遇到違反開放原始碼授權條款的情形時，並不會直接採取訴訟的手段，而會先向廠商洽詢如何取得系統程式的原始碼，除非一再的受到廠商拒絕，或是廠商遲遲不回應時，才會以訴訟為手段，由此也可以看出，開放原始碼的愛好者並不以爭訟為目的，其目的僅期望廠商均能配合開放原始碼授權條款的規範而分享其程式原始碼。

再者，對於提起訴訟的 Welte 與美國的 SFLC 來說，提起訴訟的目的並非僅為了取得金錢賠償，而是要求廠商配合提供原始碼的手段，即便要求金錢的目的也是為了要用來輔助開放原始碼運動的推展，換句話說，即便廠商希望能以給付金錢的取得開放原始碼的授權，並免除提供系統程式的原始碼責任的作法亦不可行。

與一般的商業軟體不同的是，在商業軟體公司所提出來的軟體侵權訴訟中，要求提供損害賠償的原因往往是以著作權或專利權受到侵害，但在自由軟體相關的訴訟中，由於程式開發者係以自由使用的方式進行授權，對於損害賠償的數額難以定義，因此多數的原告的訴求都是以假處分、禁制令的方式為主。這些強制的手段可以快速的阻止企業販售產品，企業也容易產生較大的損失，進而產生不敢違反自由軟體授權條款的震懾效應，是以常常被視為自由軟體侵權訴訟裡原告最常使用的手段，但是就侵權與損害賠償金額方面的判決則尚不普及。換句話說，即便廠商願意支付和解金額，也都會附帶同意遵守授權條款的承諾，廠商並不會因為支付金錢而得以免除提供原始碼的責任。

損害賠償除非法律或契約另有規定外，損害賠償的範圍應以所受損害與所失利益為

¹⁰⁹ International free and open source software law review, <http://www.ifosslr.org/ifosslr> (last visited May 13, 2010).

¹¹⁰ Skype will be open sourced, <http://linuxcrunch.com/content/skype-will-be-open-source> (last visited Dec. 27, 2009).

限¹¹¹¹¹²。開放原始碼授權條款既然以無償提供為基礎，對於所受損害所意含的積極損害，即，既存財產因損害事實之發生以致減少的情形；以及所失利益所表示的消極損害，即，因損害事實發生導致不能取得的情形而言，在請求損害賠償時便無所依據。由於開放原始碼授權條款的目的是在於讓大眾得以自由使用軟體，甚至在條文中明定不得收取軟體授權費用，因此對於損害賠償的請求與範圍往往難以界定，因此在現階段的開放原始碼授權條款的相關訴訟中，損害賠償的部份係以訴訟工本費為主。以 D-Link 的訴訟為例，原告對於 D-link 所請求的費用係以訴訟發生過程之開銷為主，在美國的 Jacobson 一案中，原告亦要求被告賠償原告因為提出訴訟而花費的基礎開銷。由於軟體在採用開放原始碼授權條款時，著作權人並不會因為軟體授權而獲得收益，因此該些與訴訟相關的費用，相較於一般商業軟體侵害著作權，所連帶要求的損害賠償之金額低了許多。

因為許多與開放原始碼授權條款相關的訴訟都採用和解方式來處理，而和解金額亦未公開，因此無法得知對廠商的金錢上的影響為何。但可以初步判斷，被告廠商願意以和解方式來處理，除了因為自知違反開放原始碼授權條外，根據開放原始碼授權條款所衍生可能使用禁制令或假處分衍生的連帶影響所發揮的恫嚇效果，才是被告廠商願意配合的主因。



¹¹¹ 民法第 216 條的規定為：

損害賠償，除法律另有規定或契約另有訂定外，應以填補債權人所受損害及所失利益為限。依通常情形，或依已定之計劃、設備或其他特別情事，可得預期之利益，視為所失利益。

¹¹² 邱聰智，新訂民法債編總則(上)，頁 267 (2003)。

第七章 實證研究分析-深度訪談

由於不同的嵌入式產品在功能、架構、規劃上，有著相當多的差異，各個嵌入式產品的開發廠商本身的研發人力、市場因素也不相同，加上相關的訴訟往往以和解收場，而不易取得足夠的判決等因素，讓本研究捨棄以量化的方式進行研究，改為利用質性研究(qualitative research)中的深度訪談(in-depth interviews) 進行實證資料的蒐集。希望能藉由質性研究的方式，探討台灣的嵌入式產品開發廠商在使用開放原始碼，進行嵌入式產品的系統開發時的經驗與想法。

深度訪談的特色為具有彈性，透過與受訪者之間的互動與交談而取得更深入的資訊¹¹³。透過深度訪談，本研究可以得知在嵌入式產品開發最前線的系統工程師，對開放原始碼授權條款的認識程度，並瞭解當前我國嵌入式產品開發廠商使用開放原始碼的情形，特別是使用 GPLv2 授權條款的 Embedded Linux 在實務上的應用，以及廠商對於開放原始碼授權條款的態度、想法。透過深度訪談的方式，不但可以發現相關議題確實值得重視，亦呈現我國產業界的作法，與可能伴隨的風險。

本研究進行深度訪談的方式為：事先擬定好訪談題目大綱，在訪談開始進行前將題目大綱提供給受訪者，讓受訪者大致瞭解將要進行討論的題目與方向，並對於訪談的目的進行說明，藉由訪談來引導受訪者陳述在利用開放原始碼進行嵌入式產品開發時的經驗，並進一步瞭解受訪者的觀點。

7.1 研究設計

本研究希望透過深度訪談的作法，得知當前我國嵌入式產品開發廠商使用開放原始碼的情形，特別是使用 GPLv2 作為授權條款的 Embedded Linux 在實務上的應用，與廠商對於相關規範的態度。本研究亦嘗試探究不同年資與產業面向之系統工程師，對於開放原始碼授權條款的認知，以及面對開放原始碼授權條款可能衍生的法律問題，或引發爭議時，我國的嵌入式產品開發廠商可能採取的作法。就訪談的題目設計而言，本研究將訪談的內容區分為五個彼此連貫，而環繞本研究重心的主題。

在進行訪談時，首先試著瞭解受訪者所任職之公司，如何進行系統程式的開發，以及系統程式的來源、不同來源之原始碼所採用的授權方式等問題，進行較為全面性的瞭

¹¹³ 石之瑜，社會科學方法新論，頁 170-172(2003)。

解；其次，詢問受訪者在參與系統程式開發時，是否使用開放原始碼，藉此瞭解開放原始碼在嵌入式產品開發時的使用情形與普及程度；接著詢問受訪者對開放原始碼授權條款的瞭解情形，與對相關法律問題的認知，藉以確認實務參與者對開放原始碼授權條款的瞭解程度；最後則詢問受訪者是否知悉與開放原始碼相關的訴訟，並假設公司的產品涉入開放原始碼授權條款之相關爭議或訴訟時，各家廠商傾向採用的應對方式等。

7.1.1 研究假設

應用在嵌入式產品中的軟體，雖然會影響消費者操作嵌入式產品時的使用觀感，但因所提供的功能相對簡易，該些軟體無法像個人電腦中的軟體般，讓一般的消費者有深刻的存在感。為此，本研究認為，基於獲利來源皆以產品的實體（硬體）銷售量來計算，以及嵌入式產品的軟體無法單獨被販售等因素，無論是晶片設計公司或是系統開發廠商在開發產品時，應會以成本低廉的開放原始碼作為首選。

除了原始碼取得成本的考量外，晶片設計公司傾向使用 GNU 所發展的開發環境來產生目的碼，此種作法有一個附帶的優點，即，對晶片設計公司而言，由於系統開發廠商同樣需要使用軟體開發環境(Software Development Kit，簡稱為 SDK)，倘若晶片設計公司使用的是商業授權版本的開發環境時，即便晶片設計公司已經支付過一筆授權費用，但對系統開發廠商而言，若要進行系統程式的修改或維護時，仍需再一次支付開發環境的授權費用。反之，若系統程式的開發環境是由開放原始碼社群所提供時，該開發環境並不需要支付授權費用，此時，晶片設計公司與系統開發廠商便不需要分開支付授權金額。

就晶片設計公司與系統開發廠商現行的合作模式而言，可以根據系統開發廠商的需求而分為兩種合作類型。在第一種合作類型中，系統開發廠商對產品的功能並無特殊的應用與需求，因此系統開發廠商會根據供應商所提供的完整方案(total solution)而生產產品，身為客戶的系統開發廠商只就操作介面上的功能，例如：顏色設定等較為簡易的設定加以變更，因此晶片設計公司需要提供的原始碼是最少的，也就是利用目的碼來提供絕大多數的程式；第二種類型則是由晶片設計公司提供主要的基礎功能，讓系統開發廠商可以根據這些基礎功能而組合出自己的產品，由於系統開發廠商需要較深入的瞭解系統程式，方能對系統程式的功能進行修改，因此晶片設計公司需要提供較多的原始碼給系統開發廠商。

另一方面，系統開發廠商與品牌廠商之間的合作模式則相對單純，品牌廠商通常不會自己改動產品的功能，而是請系統開發廠商配合修改，因此系統開發廠商並不會提供程式的原始碼給品牌廠商。即便品牌廠商欲對系統功能作進一步開發或修改時，系統開

發廠商係以目的碼搭配呼叫介面（函式）的方式來提供¹¹⁴，並不會將原始碼提供給品牌廠商。正因為品牌廠商對較少使用程式原始碼，本研究在討論廠商是否應該依照開放原始碼授權條款的規定，將系統程式的原始碼提供給終端使用者、消費者或大眾時，便不特別論及品牌廠商使用開放原始碼的情形。

根據前述的說明可以發現，嵌入式產品的系統程式原始碼，可以根據系統開發廠商對系統程式的掌握度，而分為兩種開發模式，其一為講求開發簡便而不以原始碼掌握度為要求的類型，採用此種開發模式時，絕大多數的系統程式係由晶片設計公司所掌握；其二則為希望對系統程式擁有自主權的系統開發廠商，採用此種開發模式時，晶片設計公司雖然基於雙方合作的關係而提供原始碼。不可諱言的，提供越多的原始碼給系統開發廠商時，對晶片設計公司產生的困擾也較大。因此晶片設計公司在提供原始碼時，並不會提供完整的原始碼，而且其中與晶片設計公司的核心功能相關的部分，也僅會以目的碼的方式提供，採用第二種類型的開發模式時，系統開發廠商與晶片設計公司二者，並沒有任何一方擁有完整的系統程式原始碼。

以第一種開發模式來說，晶片設計公司雖然擁有大部分的系統程式原始碼¹¹⁵，但這些原始碼的來源也不盡相同，其中若有部分透過商用授權取得的原始碼，是否可以完全遵循開放原始碼授權條款的規定，全面提供給客戶，將成為需要注意的問題。採用第二種開發模式合作時，除了同樣需要考慮原始碼的取得來源外，由於晶片設計公司與系統開發廠商均未擁有完整的原始碼。實務上，若真的要提供原始碼給消費者時，究竟應由晶片設計公司或系統開發廠商來提供，以及在提供原始碼時，又該以何種方式來提供等，這些都是操作面上可能面臨的問題。

如前所述，晶片設計公司在與系統開發廠商合作時，晶片設計公司也會避免提供完整的原始碼給系統開發廠商。對晶片設計公司來說，一旦提供原始碼給客戶，將導致使系統程式的原始碼產生分支的情形，也將連帶發生原始碼維護不同步的現象，進而使晶片設計公司與系統開發廠商在合作時產生許多問題。此外，晶片設計公司還會擔心，與自己合作的系統開發廠商同時與不同供應商合作，在其合作過程中，會因為客戶的有意或疏忽，導致自己提供給客戶的原始碼流通至競爭對手。

簡單來說，晶片設計公司與系統開發廠商均可能對開放原始碼進行改作與散布，但

¹¹⁴ 呼叫介面與目的碼的搭配使用就如同將一個功能以黑盒子的方式呈現，目的碼是黑盒子的內容，一般使用者看不出來實際的運作方式，而呼叫介面正如黑盒的輸入與輸出介面，使用者只知道在放入 A 型態的資料後，將可以由黑盒子獲得被處理過後的 B 處理結果，但其中緣由卻不會提供給使用者。

¹¹⁵ 事實上，晶片設計公司所使用的原始碼也可能有部分是透過授權而取得，無論是基於硬體設計核心而取得之控制程式，或是單純的商用授權軟體，這些由外部授權而取得的功能，很可能因授權內容的不同使晶片設計公司亦未取得原始碼。

是不同的廠商所採用的合作模式也不一定相同。隨著上下游廠商合作模式的不同，原始碼或目的碼的提供方式與範圍也不同，因此本研究認為，廠商本身的原始碼來源，以及對於提供原始碼給客戶的態度，也會影響對開放原始碼授權條款的看法，因此欲透過訪談來了解，我國的嵌入式產品廠商對於開放原始碼授權條款的態度與想法。

7.1.2 受訪對象

儘管在選擇受訪對象時，理想的方式應該是在知道整個母群及各種相關連的變項後進行抽樣並選擇受訪者¹¹⁶，但這種機率抽樣（隨機抽樣）的方式，僅適合在母群體中選取樣本機會均等時，從大範圍的母群中挑選具有代表性的樣本，並根據這些樣本的研究結果歸納出通則化的理論，進而應用至範圍更廣的母群的情況¹¹⁷。

基於各嵌入式產品開發時的封閉性與異質性，各家廠商所開發的系統不但無法進行比較，也無比較上的實益。因此本研究選擇以立意抽樣的方式，由作者挑選可提供豐富資訊的個案以進行訪談，希望選定的受訪者能夠真切反應廠商使用開放原始碼的情形。本研究在選擇受訪者時，係依據受訪對象與研究主題的強弱關係來挑選，由於晶片設計公司與系統開發廠商代表著生產嵌入式產品時的不同環節，因此受訪者中，有分別來自這兩種公司的系統工程師，也有兼具這兩種公司經驗之系統工程師，藉此得知在嵌入式產品的生產過程中，扮演角色不同時所代表的立場與思維。

本研究選擇的訪談對象包含了在晶片設計公司與系統開發廠商任職的系統工程師，並選擇不同年資的受訪者進行訪談，希望藉此得知產業歷練是否對於使用開放原始碼授權條款的態度是否一致。基於現在的開發模式多以晶片設計公司提供主要程式的作法，因此本研究中以目前任職於晶片設計公司的系統工程師為主（六位），目前任職於系統開發廠商的系統工程師為輔（兩位），而受訪者參與系統程式開發的相關年資大致分為一～五年、五～十年與超過二十年三類。

附帶一提的是，由於系統程式之開發落入營業秘密法¹¹⁸之保護範疇內，而對程式開發的管理方式，亦涉及各公司內部之研發機密，且使用開放原始碼後，各公司是否確實遵守開放原始碼授權條款之要求，而配合提供原始碼等問題，對於公司而言具有可能涉入官司或相關爭議的風險。再者，由於訪談內容可能涉及公司與員工之間所簽署的保密

¹¹⁶ Marshall, C.等著，李政賢譯，質性研究：設計與計畫撰寫，頁 90（2007）。

¹¹⁷ Derker Layder 著，簡守邦譯，社會科學研究的理論與方法：從落實理論到調適理論，頁 49（2002）。

¹¹⁸ 營業秘密法第2條規定如下：本法所稱營業秘密，係指方法、技術、製程、配方、程式、設計或其他可用於生產、銷售或經營之資訊，而符合左列要件者：一、非一般涉及該類資訊之人所知者。二、因其秘密性而具有實際或潛在之經濟價值者。三、所有人已採取合理之保密措施者。

協定與聘僱契約之保護內容，相關的技術細節必須稍加保留，因此本研究對受訪者所任職的公司與身份等資訊，均透過代碼方式來呈現。

由以下的訪談對象列表可以看出，本研究所選取的訪談對象分別任職於五家公司，其中 A、B、C 公司為晶片設計公司，D、E 公司為系統開發廠商。而受訪者代碼中，英文字母代表公司，而相同的英文字母開頭所對應的不同編號，則代表同一公司內的不同受訪者。

表 2，受訪談對象列表。

對象	工作經歷	系統開發相關年資	訪談日期
A1	晶片設計公司系統工程師	相關經驗五~十年	2010/01/01
B1	系統開發廠商系統工程師 晶片設計公司系統工程師	相關經驗五~十年	2009/12/13
B2	晶片設計公司系統工程師	相關經驗五~十年	2009/12/11
C1	晶片設計公司系統工程師	相關經驗一~五年	2009/12/19
C2	系統開發廠商系統工程師 晶片設計公司系統工程師	相關經驗五~十年	2010/01/08
C3	系統開發廠商系統工程師 晶片設計公司系統工程師	相關經驗超過二十年	2010/01/08
D1	系統開發廠商系統工程師	相關經驗一~五年	2009/12/19 2010/01/02
E1	系統開發廠商系統工程師	相關經驗五~十年	2010/01/11

資料來源：作者整理

7.2 訪談內容與歸納

以下關於訪談內容的呈現方式，將依照訪談時的問題分類分別說明，對於每個主題，首先簡要說明訪談大綱的內容與規劃提問的含意，接著以引述或簡要歸納的方式呈現受訪者的經驗與看法。

7.2.1 原始碼的取得來源、管理方式與提供方式

訪談第一部份的主題想要瞭解廠商在開發系統程式時，原始碼之取得來源、管理方式與提供給客戶的方式。訪談的題目主要有：系統程式的原始碼來源、對於不同來源的原始碼是否使用相同進的管理方式、客戶要求提供原始碼的情形、實際提供原始碼給客

戶的方式。藉以瞭解相關廠商就系統開發時的程式管理模式，以及廠商與客戶的合作模式與實際處理提供程式原始碼需求時，不同廠商所採取的因應之道。

7.2.1.1 系統程式的原始碼來源

1. 晶片設計公司

A1 表示公司的系統程式，大多是由公司的研發部門一字一句刻劃出來的，但若是可以在網路上取得免費資源的作業系統或檔案系統等程式時，便會使用開放原始碼。而 B1、B3 則說 B 公司的程式除了作業系統使用開放原始碼的 Linux 作業系統外，其他大多為自行開發而成¹¹⁹。

受訪者 C3 的回答說明了 C 公司的系統程式來源可能相當多元，他表示：

「當然方式有很多啦，那大部分，剛開始一定是先自創，先從無到有，假設系統越來越大的時候，那就會從 open source，去尋找一些合用的套件回來，有時候找不到的話，或是有需要比較高品質或是比較穩定性的軟體的話，可能就會用採購的方式，例如說一些作業系統…像這些要講求穩定高品質的東西喔，基本上是會由外購，然後來縮短開發的時程跟品管測試的時程，至於應用的話，就是應用面的程式的話，有一些是自行開發，有一些就是，比方說有一些特殊的應用的話，可能就直接買現成的…。」。

2. 系統開發廠商

受訪者 D1 表示該公司所合作的晶片設計公司，會因為晶片功能的不同而將系統程式區分為兩個部份，一部分是由晶片設計公司自己開發而成的程式，由於這個部份的程式不屬於開放原始碼，所以原始碼完全不開放。另外一個部份則是以 Linux 作業系統為基礎而開發出來的程式，上層使用的是 Android 平台的系統程式，晶片設計公司利用 Android 平台而開發一些功能。針對 Android 平台的系統程式屬於相對公開的部份，D 公司的工程師都可以自由下載這些原始碼，由於晶片設計公司的公司網頁上就有提供原始碼的下載，因此包含 D 公司之客戶或其他人也都可以自由取得、下載。

受訪者 E1 說明 E 公司所使用的原始碼主要由供應商（晶片設計公司）提供，E 公

¹¹⁹ 需注意的是，由於音訊處理的規格很多，因此通常會使用彈性較大的軟體而非硬體來處理音訊，由於音訊處理程式因為與硬體的相依性較低，也較容易利用商用授權的方式取得，所以不少公司會將音訊處理的原始碼與一般的系統程式的原始碼加以區隔。至於影像處理相關的程式，有的公司因為產品功能較為簡單而不需要提供影像處理的功能，對於提供影像播放功能的產品而言，通常因為影像處理需要較多的運算而會以硬體作為實現演算法的主體，而軟體相對需搭配硬體來設計，也較少使用開放原始碼。換言之，在系統程式中，除了影音的應用外，其餘較一般性的功能使用開放原始碼的機會較大。

司會以供應商提供的原始碼為基礎，再對系統程式作進一步的開發與改寫。但 E1 也提到，供應商願意提供原始碼的原因是建立在彼此有良好的合作關係上，事實上，E 公司也曾經因為與供應商產生不愉快，而未能取得完整的原始碼，可見晶片設計公司與系統開發廠商之間的關係也將影響原始碼的取得與否。此外，E 公司並未購買（被授權使用）商用軟體，也未使用開放原始碼。

7.2.1.2 原始碼的管理方式

無論是晶片設計公司或系統開發廠商，廠商們在進行系統程式開發時，為了管理的方便，程式通常會以分層的方式來劃分。受訪者 B1 提到 B 公司在開發系統程式時，會針對軟體堆疊的不同，決定系統工程師可以接觸的範圍。簡單來說，參與上層程式的系統工程師所能看到的原始碼範圍較小，參與下層程式開發的系統工程師所接觸的原始碼之範圍較大。

受訪者 B1:

「…應該說 driver layer 的人可以看到全部的 code, 然後 middle ware 只能看到 middle ware 以上, ap 的人只能看到 AP 以上…」。

受訪者 A1 認為，使用分層控管的方式建構整個系統的架構是為後續在維護原始碼時會比較方便，並且在考量原始碼維護的便利性起見，如果能夠不要提供原始碼給使用者時，盡量不要提供原始碼給客戶，因為會發生客戶使用先前版本的原始碼加以改動，而衍生雙方所使用的原始碼不一致的情況，在這種原始碼脫勾的情況下，若是後續要再整併功能時就會很痛苦。

有的受訪者表示，公司所開發的系統程式，亦使用了開放原始碼。至於對於原始碼的管理方式，則根據原始碼來源的不同，使用不同的管理方式。簡單來說，基本的準則是，對於花錢購入的程式，也就是經由商業授權而取得的原始碼，其管理會較嚴密。受訪者 B2 對 B 公司內部程式的管理與規劃表示：

「…架構上大概就是 AP 層，然後 middleware、然後 driver，然後最下面是 OS…，一些你跟人家有保密協定的，就只有部份的人特定的人看得到程式原始碼，那其他就開放讓人家看…」。

7.2.1.3 客戶要求提供原始碼的需求

1. 晶片設計公司

由受訪者的回答可以看出晶片設計公司認為，系統開發廠商雖然會希望取得較完整的系統程式原始碼，但身為系統程式的提供者，晶片設計公司傾向由自己提供穩定功能的程式，即，以目的碼方搭配呼叫函式的方式提供給客戶，而不需要花費太大的心力來維護兩邊的原始碼。

受訪者 B1：

「…客戶其實也是希望你的模組能夠動他就不會需要去看你的程式碼啊！除非他是剛好就是有東西要加在裡面才會需要看你的程式碼，就是人家他也不一定想看你的程式碼，大部分都是想用多少才問多少，工程師應該都這樣，如果我們底下的程式碼可以省的話就不需要提供，debug 我們自己 de 就對了啦……我倒是沒有遇過客戶一定要看我們某一支程式，我們會提供給他的會是一些 API，比如說，我們把 AP 層丟給他們，那我們 middleware 就會提供他們一些可以呼叫的 API，那 API 底下我們要怎麼改都沒有關係，只要我們可以達到他們的功能，他其實不 care…因為他們想要自己掌握的話就是 AP，AP 他們就會自己作阿，然後如果就是 AP 以下的東西有問題，我們就會解阿！…」。

受訪者 C1 表示會根據客戶所需求的功能而提供系統程式的原始碼，若客戶需要的是單純的而且很簡單的動作，就提供比較上層的原始碼，如果客戶希望作一些比較多個別的運用，才會開放比較底層的原始碼給客戶。

2. 系統開發廠商

任職於系統開發廠商的受訪者 D1，在被問到與晶片設計公司之間的合作方式時，也說明因為系統開發廠商的開發時程都比較趕，對於系統程式也不希望做太大更動，基本的架構及配置都會沿用由供應商（即，晶片設計公司）所提供的，只是視客戶的需求再作一些改變，而且晶片設計公司也會視系統開發廠商的需要與購買的晶片數量，決定是否提供部份的原始碼給系統開發廠商。

在被詢問是否提供原始碼給客戶時，受訪者 D1 表示，該公司的客戶相對較強勢，當初在洽談合作關係時，D 公司便是因為同意較其他供應商開放更多的原始碼而擊敗競爭對手，讓合作的客戶同意與 D 公司進行合作。也因此，相較於 E 公司的客戶僅要求使用者介面與目的碼的情形，D 公司的客戶會取得較多的原始碼。

7.2.1.4 提供程式的方式

1. 晶片設計公司

受訪者 C2 談到晶片設計公司的原始碼來源，以及後續提供給客戶的技術支援時，表示 C 公司的系統程式可依據原始碼的開放與否分成兩種類型，其中不對外開放的部份主要有系統程式的核心與對外採購的中間層軟體，會提供給客戶的則是上層的應用程式與使用者介面等程式。大原則就是只要牽涉到對外採購的，買進來的，通通不開放；客戶所要求的程式若與底層硬體的驅動程式相關時，C 公司在開發時就會作成是可開放式的；上層的系統程式則因為客戶通常會需要頻繁的修改，因此也會配合開放程式的原始碼。開放的部份就以原始碼的方式開放給客戶，不開放原始碼的程式，則以目的碼程式庫的方式，搭配標頭檔的函式介面提供給客戶。

2. 系統開發廠商

任職於系統廠的受訪者 D1 與 E1 在被問及公司是否提供系統程式的原始碼給客戶，以及如何提供時，兩人的回答說明了，不同公司所採用的作法也不盡相同。D 公司所開發的原始碼原則上不會提供給客戶，例外情形則是所取得的原始碼本身便是使用開放原始碼授權條款的狀況；E 公司則因為程式的原始碼若非由晶片設計公司提供，便是自行開發而得，並未使用開放原始碼，因此從未提供任何得原始碼給客戶。

7.2.1.5 小結

藉由受訪者的說明可以發現，系統開發廠商與晶片設計公司取得原始碼的目的在於開發使用，可以取得原始碼也代表廠商對產品的掌握度較高，本研究根據上述的訪談內容，將兩種類型的嵌入式產品開發廠商對原始碼來源與管理方式的作法整理如下表。

表 3，晶片設計公司與系統開發廠商之系統程式來源，與提供原始碼之方式。

	晶片設計公司	系統開發廠商
程式來源	透過不同的外部來源取得 內部發展	大多由晶片設計公司提供 小部份自行修改或調整

提供方式	外部取得的程式：以程式庫方式提供 自行開發的程式：至少會提供上層程式，並根據客戶需要提供底層程式	原則上不提供 例外： 由晶片設計公司取得的開放原始碼 （並非全部的系統程式原始碼）
------	---	--

資料來源：作者整理

7.2.2 作業系統的使用與否及其程式來源

這部份的問題希望就廠商是否使用作業系統、使用何種作業系統，以及提供作業系統程式給客戶的方式進行瞭解，藉此瞭解目前我國的嵌入式廠商使用開放原始碼之作業系統的情形，以及廠商在使用開放原始碼之作業系統前，是否考量開放原始碼授權條款的規範內容與可能形成的限制。

7.2.2.1 作業系統的使用與否

根據產品與計畫難易度的差別等因素，各受訪者使用作業系統的情形可以分為：完全不使用作業系統、使用非 Linux 作業系統，以及使用 Linux 作業系統三類。多數受訪者所參與開發的系統程式會使用作業系統，其中少數使用的是非 Linux 的作業系統。

根據受訪者所參與開發的產品不同，基本上可以歸納出一個現象，即，根據系統的複雜度不同，廠商所採取的作法亦不同。在功能需求較為簡單的嵌入式產品中，並不會使用作業系統，或使用功能較為精簡的作業系統，在嵌入式產品的功能應用較複雜時，基於時效考量，以及 Linux 作業系統的主流地位等因素，嵌入式產品的開發廠商偏向使用 Linux 作業系統。

儘管在嵌入式產品的開發過程中，作業系統的使用相當常見，但受訪者 E1 表示，自己參與的產品非但未使用作業系統，他對作業系統的評價也不高。E1 針對 Linux 作業系統被廣泛使用的情形，提出一套與眾不同的看法：

「…你如果要寫一個有效率的程式，你就不應該去用 OS，OS 是因為他去做了很多冗贅的事情…，OS 是一個非常 generic 非常 general 的一個作法，但是其實我不太喜歡這種…」。

另一方面，雖然 A 公司所使用的作業系統不是 Linux 作業系統，而是另外一套開放原始碼作業系統。但由受訪者 A1 的說明可以發現，A 公司最初在決定產品要採用何種作業系統時，不需要另外支付授權金確實為選擇時的考量。因此，對於嵌入式產品的開發廠商而言，不需要支付授權費用的開放原始碼，確實有相當程度的吸引力。

「…OS 是開發者說只要把他名字寫在跟我們產品有關的文件上，有署名的話就可以任意使用…當初在找的時候就是想說這些原始程式要可以給人家開發的，所以說如果需要授權的話，那會衍發出說我們自己要授權，給客戶時又要另外一套授權，所以當初在找的時候就是找這種免費的…」。

換句話說，晶片設計公司在發展系統程式時，也需要考量客戶未來對系統程式進行修改與發展時的授權問題。若晶片設計公司選用的是開放原始碼時，也可以避免其客戶需二度支付授權費用的情形。至於開放原始碼的使用，則因為發展模式多元，因此即便是同樣利用 Embedded Linux 作業系統的 B 公司與 C 公司，所使用的 Linux 程式來源也不相同。

B2 認為，B 公司在開發產品時的態度是，能夠使用既有的東西就不用重新自己作，以搶時效為主要考量。因此像是軟體的作業系統，與硬體的處理器核心都會選擇既有的產品，也因此成本的考量不若技術的考量來的重要。而 B 公司使用的 Linux 作業系統係由一個專屬的小部門負責取得與維護。B2 並認為商業版本的 Embedded Linux 並不一定適合自己公司的需要，因此該公司使用的是社群所開發的 Linux 原始版本，而非由 Wind River 等廠商所提供調校過後的版本。

受訪者 C3 提到，C 公司的 Linux 作業系統係透過商業版本的提供廠商而得：

「…客戶可能不會自己去編譯那個核心，因為第一個，目前雖然 Linux 是公開的原始碼，但是我們現在買的這個是有經過人家特調過的，…所以我們也不會把程式碼開放出去，只是我們會發布那個程式庫，就是我們會編譯完之後發布程式庫給客戶…」。

儘管本研究的受訪者並未經過完整的取樣，尚無法代表台灣的嵌入式產品開發廠商使用 Linux 作業系統的比率，但在受訪者所任職的五間公司中，除了一間公司並未使用系統外，其餘的四間公司中，使用 Linux 作業系統的廠商占了三間，由此也可稍微窺見 Linux 作業系統在嵌入式產品被廣泛使用的情況。

7.2.2.2 作業系統程式的取得、管理與散布方式

在被問及取得的 Linux 作業系統的原始碼，以及公司內部開發者針對開放原始碼作業系統的管理方式時，有的廠商會採用與商用的作業系統程式相同的管理方式，也就是並非所有的系統工程師，都可以看到完整的作業系統原始碼，而會控制存取者的使用權限。有的廠商會依據商用授權、開放原始碼授權的不同，對作業系統的原始碼採用不同的管理方式，對於商用授權而來的作業系統之原始碼，廠商會進行較為嚴密的管理，對參與系統程式開發的工程師分配不同的存取權限，但是對於來源為開放原始碼的作業系統，廠商的管理則相對簡單，甚至不設存取權限。

在討論作業系統係以何種方式提供給客戶時，受訪者 B1 表示，根據產品的功能與需求，B 公司的會考量不同的世代的產品所採用的作業系統，根據作業系統的來源不同，B 公司也會採用不同方式提供給客戶，針對原始碼的來源為商用授權時，B 公司僅提供使用介面與目的碼；另一方面，若作業系統的原始碼為開放原始碼時，則會大方的告知客戶所使用的 Linux 作業系統之核心程式版次，讓客戶可以自行下載。此外，B1 也說明他們在開發系統程式時，會被耳提面命要將自己所開發的新功能與 Linux 作業系統區隔開來，即，B 公司在進行系統程式之管理時，確實考量到與 Linux 作業系統相關的原始碼可能會被要求提供給後手所採取的防範措施。

7.2.3 開放原始碼之於嵌入式產品的系統程式開發

此部份的問題是想要瞭解在受訪者任職的公司中，使用開放原始碼的情形，藉此得知開放原始碼在嵌入式產品開發所扮演的角色與用途，另一方面也嘗試瞭解受訪者對於開放原始碼的認知。也就是探求廠商對於開放原始碼的起源、用途與限制的理解程度，以及身為開放原始碼的使用者時，對於開放原始碼的看法與評價，最後則與受訪者討論系統開發時，使用開放原始碼所能提供的助益。

7.2.3.1 對開放原始碼的認知

根據訪談結果可以發現，受訪者對於自由軟體的認知，通常會與共享軟體所提供的試用功能混淆，而認為可以免費使用的軟體便是自由軟體，而且受訪者普遍存在一個迷思，認為程式的開發者既然將原始碼大方的提供在網路上，使用時自然不會受到限制。換言之，受訪者的想法相當於，著作權人在提供原始碼時，便放棄了著作權法所賦予、

保障的所有權利，也就是將開放原始碼視為是公共領域的範疇，由此可以發現廠商對於開放原始碼瞭解相當不足，也可能因此而衍生風險。

受訪者 B1 認為，開放原始碼就是寫出來讓大家可以自由使用的程式，而使用者若需要使用根據這個自由軟體而改出來的原始碼時，使用者所開發出來的原始碼也應該要提供給大家使用，也就是共同開發的意思。

在搜尋引擎相當發達的今日，使用者透過網路可以取用的資源相當多，受訪者 D1 便表示：

「…就是現在網路上你如果想要什麼小東西，你打一些關鍵字進去，然後他就會已經寫好，你就可以下載下來使用，他可能就會提供什麼功能這樣子，然後如果你需要的話，就是不需要透過任何的金錢就可以…」。

受訪者 C1 認為，開放原始碼就是提供給大家，讓大家都可以任意在上面修改程式的開放性的軟體，而且軟體明確標示著不用錢的而可以自由使用，而且因為可以集眾人之力來發展，將原始碼提供給大家，如果他人有新的想法就會幫忙修改，因此程式的品質可以越改越好。

根據訪談可以發現，受訪者普遍對於開放原始碼授權條款有著很模糊的概念，但是其限制為何則明顯對其認知不足，公司關於開放原始碼的取得來源，也未特別予以規範，因此系統程式的開發者，可能知道系統程式中的部份程式屬於開放原始碼，但對於取得方式，則因為公司沒有提供一套管理制度，導致非真正採用開放原始碼的開發者，也不清楚其他人是以何種方式，由何處取得開放原始碼，甚至不知道系統程式中的哪些部份需採用開放原始碼授權條款。例如受訪者 C1 便表示：

「…就我所知，他們都說是拿 open source 啦，那 open source 到底怎麼 open 我就不知道…」。

7.2.3.2 開放原始碼之於嵌入式系統開發的用途

受訪者 B1：

「…比如說一些常用的小，一些常用的程式，比如說，之前有那個什麼，像我們會找一些測試效能的東西，或是說有時會找就是像，嗯，比如說我們要寫一個除法，就會找網路上看有沒有類似的範例程式這樣子…」。

B1 表示若參考開放原始碼時，其作法是參考開放原始碼的流程與觀念，然後自己再重新寫出一套原始碼，而這與 C 公司的作法大不相同。

受訪者 C1 表示，雖然在取得開放原始碼後，仍必須對開放原始碼稍加修改後，才能真正應用在自己的產品上，即便如此，此種作法所花費的力氣，仍比自己從無到有的開發更省事。受訪者 C1 並表示，使用開放原始碼相當於他人已經提供了組裝圖，自己只需要照著圖示來操作即可，因此會想要利用開放原始碼，以及搜尋可用之開放原始碼是一件很自然的事情，：

「…我們如果要開發一個新的功能，應該不是新功能，應該是把一個既有的東西，加在我們的產品上面的話，我們會拿，因為沒有必要去自己花時間去做這件事情…」。

對系統開發廠商而言，因為原始碼主要由供應商提供，較少對系統程式的原始碼進行改動，因此直接使用開放原始碼的情況較少，也不需要針對自己所需要的功能，而在網路上找尋開放原始碼。也就是說，系統開發廠商對於開放原始碼的需求度不如晶片設計公司來的強烈。對系統開發廠商而言，使用開放原始碼的目的，大多是為了要驗證系統程式的功能，因此使用的是完整而可執行的免費軟體，至於是否為開放原始碼則不要求。

藉由訪談可以發現，受訪者由網路上取得的軟體，可以分為附帶取得原始碼的開放原始碼，以及不提供原始碼而只有可執行格式的免費軟體。後者指的是，將個人電腦當作一個暫存該軟體的平台，利用此暫存平台執行所取得的軟體之功能，藉以輔助驗證自己所開發之程式功能是否正確，這種類型的使用因為不涉及原始碼的修改，也不會將其應用於嵌入式產品的系統程式中，因此本研究不討論廠商使用免費軟體的方法。

反觀前者，因為涉及了開放原始碼的修改與否，因此可以再針對受訪者的使用方式進一步分為兩類，一為開發者認為能利用開放原始碼來節省開發時程，而會在進行新功能開發時，習慣性的先透過搜尋引擎來尋找相關資訊，確認使否有開放原始碼提供類似功能。若有，便利用找到的開放原始碼作為開發時的基礎，接著再根據自家系統的特性，將得來的開放原始碼加以改進，例如對開放原始碼進行瘦身（刪減不必要的功能），或對開放原始碼的功能進行最佳化¹²⁰；另一種類型則是使用者雖然取得開放原始碼，但受訪者僅將開放原始碼做為參考，而於個人電腦中使用並測試，但這些開放原始碼僅做為受訪者個人的參考使用，並不會將開放原始碼整合或更新至公司的系統程式中，因此不至於發生違反開放原始碼授權條款的情形。

¹²⁰這是因為有些程式在網路上提供的時候，利用的是個人電腦系統的平台，而嵌入式系統的處理器與記憶體等元件，均無法與個人電腦上相提並論，因此原本在個人電腦上開發的程式移植至嵌入式產品時，往往因為系統資源的考量，將比較龐雜、消耗系統資源的原始碼刪除。

7.2.3.3 對於著佐權的看法

受訪者認為，以獲利為目標的商業公司，必然是以己身獲利作為考量。因此廠商若使用了具有著佐權效力的開放原始碼時，往往會相當抗拒，並採取排斥的處理方式。而這也是因為在現行產業分工的模式下，上游的供應商對於下游的客戶提供的原始碼仍有限度，遑論依據 GPLv2 授權條款的規定，對產品開發最末端的一般使用者（消費者），提供系統程式的原始碼。

受訪者 B1：

「…我覺得是一個幫助吧？你可以踩在別人的肩膀上，實際上是一種幫助吧！可是你踩在別人的肩膀上卻又不想讓人家踩在你的肩膀上，這就很奇怪，反正重點是這樣，如果人家可以幫你邁開一大步，你就可以再從那一步再往上走…」。

但是受訪者 B1 在被進一步被問到，如果因為開發嵌入式產品，而被要求提供系統程式的原始碼時，則表示不認為有必要配合 GPLv2 授權條款的規定，而提供完整的原始碼給客戶：

「…對客戶而言，他今天只要他那個功能可以用，這就是他買產品的目的阿，沒有人要買產品就要把程式碼全部都給他的吧！…」。

由此似乎可以窺見，嵌入式產品的開發廠商雖然認為，開放原始碼授權條款的要求尚屬合理，但會採取預設客戶其實並不需要程式原始碼的立場，遑論是終端的使用者。

在被問到遵守著佐權的感染效力，並問及是否會配合 GPLv2 授權條款的規定，而提供系統程式的原始碼時，受訪者均表示，一旦提供原始碼，便等同於幫助競爭對手，這也說明了廠商的顧慮。受訪者 E1 便表示：

「…基本上你把這個東西 release 出去，基本上第一件事情就是人家看你怎麼寫然後就直接抄…」。

7.2.3.4 對開放原始碼不提供擔保的看法

儘管這部份的問題並不在事先準備的訪談大綱中，但由受訪者在訪談中所提到的這些想法也可以看出，嵌入式產品開發廠商在使用開放原始碼時的一些考量。舉例來說，受訪者 B1 認為開放原始碼的麻煩在於，沒有辦法取得需要的支援服務，因為開放原始

碼是不用錢的，一旦開放原始碼的功能有問題的話，也要自己去找才找的出來並解決。因此 B1 表示，在決定是否使用開放原始碼前，會先判斷開放原始碼的成熟度，如果夠成熟的話必定會是選擇使用開放原始碼，因為夠成熟的話代表問題會比較少，如果開放原始碼的成熟度不夠的話，則寧可自己重新撰寫、開發程式。

受訪者 C3 也表示，在需要品質要求比較高的考量下，會選擇使用商用軟體，由此也可以看出，嵌入式產品開發廠商對於開放原始碼的品質不具信心，而抱持著品質堪虞的不安心態。這部份的反應則與一般開放原始碼開發者認為，集眾人之力而開發的開放原始碼，會因為在更多人監督下發展，而能提供較佳品質的思維截然不同，這可能也是因為嵌入式產品的異質性較高，社群在發展開放原始碼計畫時，無法像在個人電腦一樣以相同的開發平台作為發展基礎，進而使合作的效果較為分散所導致。

7.2.4 系統程式的原始碼管理與軟體授權相關的法律問題

此部份的主題是希望瞭解各公司以及程式開發人員，對於系統程式在開發過程中，與軟體授權相關的法律問題是否重視，以及關切的程度。各家公司對於系統開發時的相關法律保障情形不一，制度較為完善的公司會在開放原始碼導入前，由法務部門進行瞭解，並在系統程式開發過程中耳提面命，提醒參與開發的系統工程師注意相關的程式使用方式，另外還會有負責把關的人選來進行作業系統來源程式的管理，且有意購買提供授權條款檢測功能的管理軟體。

B 公司是受訪者所任職的公司中，極少數會較為注意相關措施的廠商，除了受訪者 B2 表示該公司有考慮購入確認授權模式的管理軟體外，受訪者 B1 也表示：

「…我們在用 Linux 這個開放原始碼的系統時都有先上過一些課程，就是會開一些課程告訴我們說盡量要用什麼樣的方式來使用會比較好，那如果對模糊地帶的話，也會講說要問一下誰，比如說問主管，或是問一些像是我們有法律顧問…之後我們會利用一些，比如說外面販售的軟體去 trace 說我們的 code 到底是不是有用到，這樣的事情，作 code 的把關…」。

嵌入式產品開發廠商之所以會自行開發硬體驅動程式，是為了要控制自家產品的硬體功能，因此無法全部採用外來的原始碼。舉例來說，產品的控制晶片可能針對硬碟存取的功能提供控制，因此嵌入式產品的開發廠商會針對自家的晶片功能撰寫，但是因為硬碟的資料傳輸介面是產業界常用的標準，因此上層的存取函式已經有現成的開放原始碼可以使用，在時效的考量下，嵌入式產品的開發廠商只需要針對下層的硬體進行控制程式撰寫，並提供上層的存取函式使用即可。

受訪者 B2：

「…像我們都是用自己的硬體功能去實現，所以等於你的驅動程式是以那些東西為基礎而發展的，那些硬體功能的東西我覺得比較不會有侵權的問題啦…」。

需注意的是，即便下層的硬體控制程式係由受訪者自己撰寫，但受訪者認定自己寫的程式就不會有問題的心態其實仍待商榷，因為依照 GPLv2 授權條款的規定，GPLv2 授權條款對於衍生著作也具效力。假設上層的應用程式是採用 GPLv2 授權條款的開放原始碼時，代表廠商自行開發的下層驅動程式，仍可能受到上層程式的感染。因此除了上層原本即為開放原始碼外，下層的驅動程式也可能受到 GPLv2 授權條款的著佐權效力影響，而需要一併提供其原始碼。也就是說，系統程式開發者的這種認知實際上仍可能造成違反 GPLv2 授權條款的可能與風險。

至於在系統開發廠商任職的受訪者 D1 表示，該公司的系統程式原始碼係由供應商提供，之後再進行小幅度的功能修改，並未思考過該公司現行的原始碼取得來源，與系統程式的開發模式是否可能衍生法律風險，似乎可以看出受訪者認為自己的程式修改是在既有授權下所為，便不至於衍生問題才是。

受訪者 D1：

「…我覺得因為我們是去修改，所以整個架構其實是人家的，那人家的話，你給他就是稍微改一下的話就是，嗯，從紅色改成藍色之類的，其實應該是很保險的，不會有任何法律問題，呵呵，就是，其實我們沒有想過啦…」。

在程式來源係採用商用授權模式時，原始碼的取得固然是經過授權而可以適用，但是類似的思維對於開放原始碼而言，卻不一定可以直接適用。相較於使用原始碼時的授權，GPLv2 授權條款更關注程式在散布（販售產品）時，嵌入式產品的開發廠商應配合提供原始碼給後手的規定。

另外，藉由受訪者 E1 的答案也可以看出，研發部門對於法律議題相當陌生，也不甚關心：

「…哪有時間去想這種問題，法律問題大概也不是我的…」。

對於實際參與系統開發的工程師而言，即便是來自供應商的原始碼，亦即，透過商用授權而來的軟體，系統工程師也不一定清楚不同的授權條款彼此間的區別，受訪者 D1 表示：

「…我知道他的文件上也會，然後像他就會寫密密麻麻一大堆，主要的就是不能去

分享或是去複製用來幹麼，坦白說，我們使用者不會太去關心這個，那你大概知道他的意思就是希望你不要去隨便給別人他的東西…」。

由訪談的過程可以發現，除了 B 公司外，其他的公司並不重視使用開放原始碼時，可能衍生的相關法律議題。儘管多數受訪者表示公司內部會提供教育訓練，藉以加強工程師關於智慧財產權的認識與瞭解，但在筆者進一步詢問其課程內容後，可以發現該些公司所提供的訓練課程多著重在專利方面，且工程師對這些課程的印象均相當模糊。換言之，對於研發人員來說，開放原始碼應該如何正確的被使用，以及開放原始碼授權條款的規範，是否成爲公司產品在銷售時的潛在法律問題，往往抱持著不知道也不想關心的態度。

這些前線人員不關心法律議題的情形，可能是因爲一般公司內部的跨部門互動不足，例如：除非研發主管在自覺有疑義時，才會考慮尋求法務部門的協助，但是這樣的前提是研發主管對於法律問題有很敏銳的觀察力；或是因爲購買商用軟體時，由於公司規範的採購流程中，規定採購過程需要法務部門簽核時，才會產生跨部門的互動。但是開放原始碼卻因爲太過容易取得，系統工程師可能自行透過上網搜尋而取得，並自己應用到系統程式中，亦即，系統工程師在使用搜尋而得的原始碼時，並不一定會知會研發部門的主管，遑論跨部門通知法務部門。正因爲開放原始碼因爲太容易取得，而使用開放原始碼的工程師又不一定意識到使用開放原始碼後，對於公司產品可能造成的影響，對於公司而言，等同於一個潛在而未知大小的風險。

7.2.5 使用開放原始碼的法律爭議

這部份的訪談題目首先詢問受訪者是否清楚不同類型的開放原始碼授權條款的拘束性，以及開放原始碼授權條款的規範內容，是否會對各公司在進行系統程式的原始碼開發與管理時的困擾。並希望瞭解我國的嵌入式產品開發廠商，是否關心開放原始碼授權條款所引發的爭議與訴訟等，以及該些廠商是否知道相關的判決結果，並詢問受訪者一旦自家公司的產品遇到被開放原始碼的著作權人主張權利，並要求提供系統程式的原始碼時，受訪者會傾向以何種方式處理。

根據受訪者的反應可以看出，受訪者對於著佐權效力略有聽聞，但不清楚各種開放原始碼授權條款間存在許多差別，甚至誤認爲不同種類的開放原始碼授權條款可以提供廠商在使用原始碼的時候進行選擇，例如在使用了 Linux 作業系統後，將系統程式以 BSD 授權條款的方式來授權。

在進一步被問及對開放原始碼授權條款相關訴訟的瞭解程度時，僅有少數的受訪者

對於相關的爭訟有所聽聞，大多數的受訪者則渾然不知，對於相關訴訟的內容能夠瞭解的更是所剩無幾。此外，對於 GPLv2 授權條款的強大感染力，導致訴訟結果一面倒的情況，受訪者普遍認為 GPLv2 授權條款的規定，對於以營利為目的的廠商而言過於嚴苛。

受訪者 A1 便認為：

「…當然如果授權沒看清楚的話就用下去的話，在法律上判決當然是合理的，可是實際上，說真的，實際上我是覺得還蠻不合理的…」。

另外，受訪者 E1 也認為，即便現有的一些商業公司的軟體，選擇使用開放原始碼授權條款，但那些公司的動機並非分享其產品，而是因為錯失市場上佔有率，乾脆利用開放原始碼授權條款來提高產品的使用率，並藉此打擊對手。亦即，即便商業軟體公司選擇使用開放原始碼授權條款，但其目的僅是用開放原始碼來作為公司產品促銷的手段，如果商業軟體公司仍有選擇時，開放原始碼授權條款自然不會是商業軟體公司所樂於選擇使用的。

接著，受訪者被問及假設性的問題，即，倘若公司使用了開放原始碼而未依照規定提供原始碼，在被開發社群發現後，被要求該公司遵守開放原始碼授權條款，例如：依照 GPLv2 授權條款的「相同方式分享」規定，配合提供公司產品的系統程式原始碼時，是否會配合提供原始碼。受訪者 B2 認為此種要求對於公司而言相當棘手。因為公司傾向不提供原始碼，甚至樂於付費使用，對於這種不要求給付授權費，而只要求提供原始碼的情形感到很困擾。

更進一步假設，若受訪者任職公司的產品，因為未提供系統程式的原始碼而引發訴訟時，受訪者 B2 會選擇放棄使用現成的開放原始碼，他認為提供原始碼會對公司造成很大的衝擊，即便重新自行開發原始碼較為費時，仍傾向選擇自行開發類似功能的軟體，藉此也可以推知，提供系統程式的原始碼給後手，對於嵌入式產品開發廠商的機密揭露有很大的影響。

部份的受訪者則認為，公司在販售產品時，仍應以時效作為主要考量，倘若被要求遵守授權條款的規定、提供系統程式的原始碼時，他們願意配合其規定，提供產品的系統程式原始碼。受訪者 B1 便表示，公司的立場應該還是以搶時效為主，所以會繼續選擇使用開放原始碼，並配合提供系統程式的原始碼。他的想法是，系統程式的架構與內容相當繁雜，即便是同一個公司的同仁，在相對具有背景知識的情況下，也不見得能夠通盤了解系統程式的運作，更何況是完全不相干的外人。亦即，系統程式原始碼的提供，並不同於瞭解系統程式的運作。受訪者 C1 也有類似的看法，受訪者 C1 便表示：

「…反正把 code 開放，大家也看不懂，沒關係，因為自己都看不懂了，不用怕…」。

受訪者 C2 表示，在過去的經驗中，系統程式的來源係以商業授權與自行開發為主，較少接觸開放原始碼，無法明確的回答這個假設性問題。此外，根據受訪者 C3 的回答，似乎可以窺見，C 公司仍採用以商業授權方式來避免爭訟的想法，卻忽略了開放原始碼重視分享更甚於授權金的本質與思維。

「…事實上我們現在已經有用買的，而且我們也一直在預防被告，所以理論上這個假設的情況應該是不會發生的，因為我們已經做好了預防動作了，我們怕被告的部份我們盡量防，我們覺得沒把握的我們盡量買…」。

藉由上述訪談可以發現，各公司內部在管理系統程式的原始碼時，鮮少注意開放原始碼授權條款的規定，也未考量在此種規範下，應該如何在系統程式開發流程中控管，這很可能是因為對開放原始碼授權條款的不瞭解，甚至誤解而產生的。

在開放原始碼的相關訴訟日益增加的今日，台灣的嵌入式產品開發廠商若在系統程式中，使用了開放原始碼，究竟是因為在評估過風險後而作的決定，還是因為不清楚也不願意瞭解開放原始碼授權條款的規定，目前尚未能妄下定論。然而，一旦發生相關訴訟時，廠商亦不能因為對開放原始碼授權條款的規定不知情，而免除其提供原始碼的責任，因此廠商的作法也有檢討的必要。

7.3 小結

在開放原始碼普及發展下，本研究最初假設廠商在選擇系統程式的開發方式時，應會以成本低廉的開放原始碼作為首要考量，但由訪談結果可以發現，廠商選擇使用開放原始碼的原因，不全然是因為成本較低的關係，而會考量穩定性等其他因素。有些時候，廠商在開發嵌入式產品時，是迫於 Linux 作業系統的廣大市佔率或因為客戶的要求，而不得不配合使用 Linux 作業系統，部分的廠商甚至希望可以利用付費的方式，獲取開放原始碼而不需提供原始碼，只是苦無其門而入。

其次，根據訪談內容可以發現，無論廠商在使用開放原始碼時之動機為何，多數廠商對於使用開放原始碼時，可能伴隨的相關規範之注意程度，較一般商業授權軟體之注意程度更低，這部分的訪談內容與本研究的假設相符。

另一方面，本研究在進行訪談前，假設我國廠商混雜使用不同來源的原始碼，而未針對開放原始碼的部分予以區隔，但在訪談之後發現，已經有部份的嵌入式產品開發廠

商注意到使用開放原始碼時可能產生的限制，該些廠商也逐步將系統程式的原始碼加以切割，並規劃使用授權管理軟體，顯示我國廠商就此方面之注意程度其實較本研究最初的假設更充分。

再者，儘管一般人認為開放原始碼透過社群的協助而能有較佳且穩定的品質，但由訪談結果可以發現在實際應用時，廠商對開放原始碼的品質感到懷疑，這可能是受到各家產品先天設計不同，導致許多功能會因應廠商的硬體設計而異，在一般使用者（消費者）無法得知該些差異的情況下，開放原始碼社群所能提供的資源，將無法如個人電腦上的軟體開發一樣達到匯集的效果。

關於使用開放原始碼可能衍生的法律爭議，訪談結果與本研究最初的假設相同，即，我國廠商往往埋頭於產品開發，而未特別考慮到法律層面的問題，對於相關的爭議與爭訟也不清楚。網路的普及固然使開放原始碼的使用更為便利，但就另外一個層面來看，廠商在進行產品開發時，也可能因此而更容易在不知不覺中，違反了開放原始碼授權條款的規定，訪談結果不但證實國內的嵌入式產品開發廠商，對於開放原始碼授權條款的瞭解仍然不足，也應證了本研究對於廠商抱持得過且過的心態的假設。



第八章 建議、結論與未來研究方向

8.1 建議與結論

根據 Linux Foundation 在 2008 年所作的報告顯示¹²¹，如果 Linux 的核心程式(Linux Kernel)所採用的開發模式是由廠商獨自開發而非現在的社群主動開發模式，單就開發 Linux 核心程式便需要耗費美金\$1.4 billion，而整個 Linux 作業系統的原始碼更需要高達美金\$108 億¹²²的花費方得完成。換句話說，Linux 作業系統的開發，若非透過開放原始碼的開發社群來一起達成，而是以一般私人公司開發程式的模式，爲了達到目前 Linux 作業系統原始碼的規模，所需付出的開發成本相當高，也鮮少有公司能夠獨立負擔此種成本。換句話說，廠商得以利用 Embedded Linux 發展系統程式都是架構在這些基礎才得以發展的，也因此 GPLv2 授權條款的規範雖然較爲嚴苛，卻也確實促進了 Linux 作業系統的蓬勃成長。

隨著開放原始碼運動的推展，近年來，如何利用開放原始碼作爲商業使用的討論^{123,124}亦應運而生，基於產品本質的先天差異，不同類型的產品所適合的商業發展方式與開發模式自然也不同。舉例來說，就嵌入式產品的硬體製造商而言，因爲以硬體作爲販售標的，因此能夠利用發展雙重版本¹²⁵，或是提供軟體服務的方式而獲利的可行性便較低，即便與同樣以硬體作爲販售來源的伺服器廠商¹²⁶相較，在嵌入式產品單價遠低於伺服器產品以及更新、淘汰速度相對較快速的的情況下，嵌入式產品的開發廠商也無法直接採用類似伺服器產品開發廠商所使用的商業模式。

本研究認爲，若能在兼顧 GPLv2 授權條款的分享理念，與保障廠商利益之間取得平衡，不但可以維持既有關於開放原始碼的交流與成長模式，也可以更進一步吸引廠商加入，將更多的資源導入至開放原始碼的交流、創新與開發，進而發揮更大的效益。

8.1.1 對 GPLv2 授權條款的修改建議

儘管目前 Embedded Linux 使用 GPLv2 授權條款進行授權爲一個既定事實，但 Torvalds 最初選用 GPLv2 作爲 Linux 作業系統的授權條款時，Linux 作業系統係以個人電腦作爲

¹²¹ Amanda Mc Pherson et., Estimating the Total Development Cost of a Linux Distribution <http://www.linuxfoundation.org/publications/estimatinglinux.php> (last visited Jun. 6, 2010).

¹²² 根據 2008/5/13 所公佈的 Fedora 9, linux-2.6.25.i686 版本所進行的估算。

¹²³ See WEBER, supra note 61, at 190-223.

¹²⁴ See ROSEN, supra note 21, at 262.

¹²⁵ 例如：MySQL、Red Hat、Mozilla。

¹²⁶ 例如：IBM、SUN、HP。

操作平台，而今的時空背景與當時並不相同，嵌入式產品的軟體發展與個人電腦的軟體發展在本質有著諸多不同，讓廠商無法採用與個人電腦平台相同的模式來獲利。

再者，根據 Linux Kernel 的使用說明，可以看出 Linux Kernel 的開發人員對於嵌入式產品使用 Linux 作業系統，卻未遵守 GPLv2 授權條款而提供、分享其原始碼的情形並非不知情，也知道在嵌入式產品的應用時，廠商必須再投注許多心力，將 Linux 作業系統的核心程式中，關於硬體控制的部份進行修改，也就是針對自家的產品進行量身打造。換句話說，嵌入式產品與個人電腦廠商在使用開放原始碼時，從中獲得的效益確實不同，因此本研究認為，GPLv2 授權條款對於原始碼開放的要求程度，似乎也應該針對被授權人的實際獲益程度而有不同的考量。

另一方面，Linux 作業系統的核心程式開發者既然認為，當嵌入式產品廠商選擇使用 Linux 主線¹²⁷的程式發展時，該廠商的產品品質必然較佳，進而吸引一般使用者（消費者）購買遵守 GPLv2 授權條款的廠商所開發的產品時，是否認為市場存在著選擇機制來自動淘汰不遵守 GPLv2 授權條款的廠商？在這樣的基礎上，似乎代表開放原始碼的推動者，可以利用市場的選擇機制，來壓抑不配合開放原始碼授權條款者的產品銷售，而形成一種利用消費者力量支持願意分享創新的廠商。

附帶一提的是，儘管 GPLv3 授權條款在 2007 年推出，但是將 GPLv2 授權條款與 GPLv3 授權條款互相比較時，可以發現，GPLv3 授權條款不但保有著佐權的特性，甚至有著更多限制更多。例如：為了防止類似 Tivo 的作法，也就是避免被授權人利用 DRM 手段，以硬體的方式限制使用者，讓使用者無法在其產品上執行經過修改的軟體，進而讓軟體自由的目的無法達成，GPLv3 授權條款便特別防範類似的行為¹²⁸。換言之，由 GPLv3 授權條款的規定可以看出，目前社群的主張與廠商的需求完全是背道而馳的，在雙方無法得到共識的情況下，當廠商真的使用開放原始碼時，自然不樂意落實開放原始碼授權條款所賦予的責任，而會以偷偷摸摸的方式來使用。

承上，與其讓嵌入式產品的開發廠商對 GPLv2 授權條款抱持敬謝不敏的態度，而以私下使用的方式，對於開放原始碼授權條款的制定者而言，這種讓廠商以防範的態度，而非樂於參與的規定是否是最好的方式？在這些考量下，本研究認為有必要特別針對廠商在開發嵌入式產品時的需求，與開放原始碼授權條款的規範進行探討。

事實上，若是開放原始碼授權條款的規定，可以考慮到執行平台的本質差異，並將嵌入式產品與一般性的個人電腦之軟體程式的不同列入考量，以較為可親的方式規範原

¹²⁷ Linux 的開發將主要進行的計畫稱為主線（mainline），而開發過程因為分支（fork）而產生的程式碼則稱為分支核心程式（out-of-tree kernel code）。

¹²⁸ 例如：GPLv3 授權條款的前言與第六條。

始碼的分享方式，說不定能夠促使廠商更樂意投入原始碼分享的行列。

由於目前的 GPLv2 授權條款中，著佐權的影響層面過大，讓使用 Linux 作業系統的廠商心生畏懼，但迫於開發時程或 Linux 作業系統的主流地位，導致嵌入式產品的開發廠商以偷偷使用的方式，這種透過嚴格規定卻讓被授權人（嵌入式產品的開發廠商）膽怯的方式，並未使原始碼的流通達到最大的效益。本研究認為 GPLv2 授權條款若能對嵌入式產品應用的情形，採用較為務實的作法，反而可以讓廠商更願意投身於原始碼的分享，因此本研究嘗試對 GPLv2 條文中對於嵌入式產品應用較為限制的部分提出修改的建議。

追根究底，GPLv2 授權條款的感染性令廠商感到相當畏懼，對於嵌入式產品的開發廠商而言，若是採用類似 Artistic 授權條款的要求而標示原始碼的來源，雖然仍可能有所顧忌而不會確實遵守，但是提供原始碼來源相較於提供完整原始碼的內容而言，對於廠商的影響相對較低，對於機密外露的機會也較少，相信這樣的授權條款規定對廠商是相對容易配合的作法。

另一方面，對開放原始碼授權條款的推動者而言，讓程式的原始碼得以公開與分享係其致力推動的最終目的，因此如果廠商僅公布其原始碼的來源，卻未公布修改後的原始碼，卻未能得知其細節，對於自由軟體的推動無異於隔靴搔癢。

在考量兩者的訴求與利益下，本研究認為可設定一個原始碼暫時保留而不公開的期間，讓廠商可以在一段固定期間內，保留其系統程式開發的隱密性而不需公開其原始碼，但是在該段期間經過後，則要求廠商提供系統程式的原始碼，如此便可以兼顧廠商的權益與社群的訴求。

事實上，嵌入式產品的系統程式開發，特別是就底層程式而言，原始碼的內容會依據公司所使用的硬體而異，在一般使用者、消費者不瞭解硬體的設計時，即便廠商提供了原始碼，但一般使用者、消費者也無法將相同的原始碼應用至其他種類的裝置上。亦即，由於嵌入式產品的硬體並不相同，廠商所提供的原始碼無法如個人電腦一樣，適用於所有的同類型產品，例如：Apple 的 iPhone 便無法執行 HTC 手機的系統程式。反過來說，這也是嵌入式產品開發廠商認為，網路上的開放原始碼雖然提供了開發的基礎，卻無法直接套用的原因，廠商既然需花費心力加以改寫原始碼，便不應該被要求提供原始碼，因為他們的成果並非單純的不勞而獲。

承上所述，單純的提供系統程式的原始碼，並不同於可以實際利用原始碼 (ready to use)，因此要求廠商制式化的提供所有的原始碼時，對於程式開發者間的交流與互動等的助益其實相當有限。對於廠商而言，更重要的是原始碼背後所代表的演算法、觀念

與技巧，因此，對於共通部份的原始碼予以分享才是真正能促進交流的作法。

在不要求廠商提供全部的原始碼的情況下，廠商只有在其原始碼的來源為開放原始碼時，該部份的原始碼在經過改作後，仍需提供修改後的原始碼，對於自行開發的原始碼，則因較具有特殊性與產品特性等因素，對於一般使用者的幫助其實相當有限，關於這部份的原始碼就不需要強制規定廠商應該提供。本研究以為，最初分享原始碼的作者固然可以決定自己所開發的原始碼要被分享給他人，但是原作者的權利不應該擴及改作者的著作。讓一般使用者可以就原本的開放原始碼加以改進，並對針對該部份的改進與他人進行交流，而不強求嵌入式產品的開發廠商所有的原始碼，都必須以相同的開放原始碼授權條款來授權時，此種作法不但不會對其他來源的原始碼造成限制，反而可以讓廠商更願意配合。

換句話說，就嵌入式產品而言，若程式的原始碼採用的是 MPL 類型之開放原始碼授權條款時，其普及程度將大幅提升，或是如 Linux 作業系統的原作者一樣，自願放棄、限縮在 GPLv2 授權條款所賦予的部份權利時，其原始碼的普及程度才能更加廣泛。

易言之，與其讓開放原始碼授權條款的效力過於強大而讓廠商心聲畏懼，並不會使廠商放棄使用開放原始碼，而只是加深廠商因為擔心使用開放原始碼的情形被揭露而對自身原始碼進行嚴密的保護，因此本研究建議 GPLv2 授權條款可以針對不同的應用而給予提供原始碼的不同期限規定。就嵌入式產品的應用來說，這類型產品的生命週期並不長，因此如果讓嵌入式產品廠商在一段時間內，得以因為產品的販售而獲利，且在該段期間內可以不必提供程式的原始碼，但在該段期間過後，廠商便應該提供系統程式的原始碼。如此一來，廠商在該段受到保障的期間毋須擔心提供原始碼造成競爭對手的仿效，但在該段期間過後，開放原始碼授權條款的效力仍然得以發揮。

本研究認為此種設定一段期間後才要求提供原始碼的作法，是一種兼顧廠商權益且不違背自由軟體分享本意的作法。事實上，由於嵌入式產品廠商在開發系統程式時，往往以累進的方式在發展，也就是在既有的架構上發展新功能。如果廠商將舊產品的原始碼分享後，能有公司外的其他使用者提出修改建議時，廠商亦可參酌其建議而應用其修改於新產品中，在提供廠商一定之獲利基礎的情況下，輔以外部開發支援之誘因，讓開放原始碼授權條款的效力得以被確保，而不是單純以恫嚇的方式來威脅廠商，可說是一種提供誘因讓廠商樂於配合遵守開放原始碼授權條款的方式。考量嵌入式產品的生命週期下，保障不提供原始碼的期限可以不用被設的太長，例如，以產品上市後的兩年作為一個分水嶺，當然，詳細的期間亦可能因為產品的生命週期不同，而應給予不同的範圍。

當資源可以被自由使用時，舊的秩序才會被打破並發生改變，因此，創新不應該受

到限制¹²⁹。即便 GPLv2 授權條款是以強制分享為基礎而希望讓軟體的開發能夠更自由，但是 GPLv2 授權條款本身又未嘗不是一種限制，但是過於強烈的限制對後續的被授權人（嵌入式產品的開發廠商）而言卻容易成為望而生畏的枷鎖，嵌入式產品的開發廠商若不希望違反 GPLv2 授權條款的規定，則可能導致嵌入式產品的開發廠商完全避免使用 GPLv2 授權的程式，讓嵌入式產品的開發廠商無法將其他授權的程式合併於其中，或者嵌入式產品的開發廠商只能暗中進行，默默的違反 GPLv2 授權條款，暗自期望自己不會成為自由軟體提倡者提告的對象，這樣一來，GPLv2 授權條款反而成為軟體發展上的限制或負擔，卻未發揮鼓勵創新的效果。

綜上所述，在搭配上上述的開放期間、分層開放與分檔案開放等手段，除了可以避免因為授權條款之間的衝突，導致在實現原始碼之分享時的問題外，也讓廠商得以在更大的誘因下，願意配合開放原始碼授權條款的規定，因此可以更促進開放原始碼的效果。舉例來說，規定在產品販售的一年內，廠商得以不提供系統程式的原始碼；在產品開始上市並販售後的一至兩年期間則必須採用分層的方式開放，在兩年之後則必須提供系統程式的完整原始碼，即，使用 GPLv2 授權條款的上層應用程式在第一年至第二年的期間可以不提供程式原始碼，但是在第二年之後就必須要被強制提供。

本研究嘗試就兼顧雙方立場的作法提出對 GPLv2 授權條款的修改方向建議，並認為 BSD 類型授權條款所採用的全面自由使用的精神會讓程式開發的效果更加多元，也讓創新得以不受限制，至於實際上對於 GPLv2 授權條款的修改細節，例如提供原始碼期間應該如何規定，是否區分提供原始碼的階段（程度）等，都可以再針對嵌入式產品的特性進行更進一步的討論。

8.1.2 根據現行 GPLv2 授權條款給予廠商的建議

根據前述訪談的結果可以發現，目前國內廠商對於開放原始碼授權條款的效力仍不重視，雖然政府機關與非營利團體也對開放原始碼的概念加以推廣與提倡，但就嵌入式產品開發廠商而言，需要考量的面向也較多，因此往往會抱持著駝鳥心態而裹足不前。針對嵌入式系統製造商的難處，本研究提出以下建議，期能協助廠商在遵守相關的開放原始碼授權條款的情況下，以最不受到影響的方式提供原始碼，進而達到兩全其美的效果。

簡要來說，本研究給予廠商因應開放原始碼授權條款所應採用的作法如下：首先，各公司應結合公司的內部資源並正視開放原始碼授權條款問題；其次，在不同授權模式

¹²⁹ See Lawrence Lessig, *The Architecture of Innovation*, Vol. 51, Duke L. J., 1783, 1801 (2002).

同時存在的情況下，公司應考慮使用雙重授權的軟體¹³⁰；再者，公司內部在發展系統程式時，應該對程式的開發過程進行相關的防護措施；最後，本研究建議各公司仍應遵守開放原始碼授權的條款，在對於公司影響最小的情況下，配合提供系統程式的原始碼。

8.1.2.1 結合公司的內部資源並正視開放原始碼授權條款問題

儘管開放原始碼授權條款的討論已較過去普及，但我國廠商在相當依賴開放原始碼進行嵌入式產品開發的同時時，卻對開放原始碼授權條款的相關議題不甚瞭解。絕大多數的系統開發者對於自由軟體與開放原始碼的了解，仍處於曾經聽聞卻不了解的狀態，即便曾經看過相關的開放原始碼授權條款的名稱，卻鮮少會進一步探究該些開放原始碼授權條款的規範內容，對於開放原始碼授權條款所引發的相關爭議或訴訟的了解更是少見。

廠商在發展嵌入式產品的功能時，系統程式的開發與選用，往往由研發部門根據產品功能的需要而決定，但一般的程式開發者並沒有能力對開放原始碼授權條款進行正確的解讀與判定，因此本研究建議公司的法務部門提供研發部門相關的諮詢與協助。再者，由於開放原始碼的使用對於公司法務而言亦可能是一個不熟悉的領域，因此由外部的專業公司¹³¹所提供的諮詢服務也應運而生，若公司的智權法務部門對於各式各樣的開放原始碼授權條款規定不甚熟悉時，該些業者也可以輔助各公司關於開放原始碼使用制度的建立。

如此一來，公司內部應可建立出一套判斷準則¹³²，對於開放原始碼的使用與否提供給研發部門作參考，亦即，研發部門不應該是判斷開放原始碼是否被使用的唯一判斷部門，應該將開放原始碼視同一般的商業授權軟體，藉由跨部門的諮詢做出最終的判斷。

8.1.2.2 使用雙重授權的軟體

對於把提供原始碼視為大忌，且願意支付授權費用的廠商而言，在開始採用開放原始碼授權條款的程式前，可以先找尋類似的功能是否存在採用商業授權的其他版本，亦即，該軟體程式是否以雙重授權方式存在，若是，便可以採用其中的商業授權方式來取得軟體。如此一來，廠商雖然必須付出軟體授權費用，但是其成本是可以確切估算的，

¹³⁰ 例如：對 Sun 的 Star Office 與 OpenOffice，以及對 eCos 與 eCos Centric 的選用。

¹³¹ Brain League, Open Source License Services, http://brainleague.com/offering/open_source.htm (last visited Jun. 5, 2010).

¹³² 請參見附錄二。

相較之下，廠商在提供原始碼所衍生的風險大小卻是未知的。

1. Linux 作業系統

對於使用 Linux 作業系統的廠商而言，由於 MontaVista 與 Wind River 等廠商以提供商業版的 Embedded Linux 的技術服務為主，這些公司對 GPLv2 授權條款的內容也知之甚詳，若能藉由這些公司的協助，對於如何在不違反授權條款規定的使用下提供對公司利益的保護必然可以提供較多的參酌，透過這些公司所提供的服務，對於我國的嵌入式產品廠商而言，不僅研發時可以享用便捷的技術資源，在使用原始碼時，也多了一道法律層面的屏障。

儘管廠商使用商業版本的 Embedded Linux，代表廠商仍然必須付出相關的軟體使用費用，但採用此種授權類型時的授權費用，仍較完全使用個別開發的商業軟體版本來得低，且相關的程式仍然可以獲得，而仍有其利基存在。再者，商業授權版本的 Embedded Linux 與完全免費即可取得的 Embedded Linux 版本之開發成本，並不是單就原始碼的取得成本，便能推得整體開發的成本。因為各家廠商的硬體設計不盡相同，用來控制硬體的作法（程式）也各異，因此廠商對於網路上由社群開發的 Linux 作業系統之原始碼時，都必須作進一步的修改，方能實際應用至產品中，而進行這些修改所需的人力與時間也代表廠商所負擔之額外的成本。易言之，前者雖然需要付出代價才能取得，但在考量取得程式後，廠商所需進行的維護與使用所需的成本後，前者所花費的整體開發成本卻不一定較後者高。

Red Hat 所推出的 Linux 作業系統，在個人電腦與工作站平台的市場上佔有一席之地，Red Hat 在該些平台所採取的開發模式是，參與 Linux 作業系統的改良，並公開改良過的 Linux 作業系統的原始碼，讓投身於 Linux 作業系統開發的社群可以對該版本進行測試、回報甚至修改，而 Red Hat 便可以利用社群的資源，作為 Linux 作業系統改進的依據，之後 Red Hat 再將效能趨於穩定的原始碼，進一步更新到另一套需商業授權的版本¹³³中，讓被授權人可以使用經過優化後，品質較為穩定之 Linux 作業系統。

2. eCos 作業系統

此種發行測試版本後，再另行提供商業版本的類似作法，也被 Red Hat 應用在嵌入式產品的作業系統中。針對嵌入式產品的部份，Red Hat 曾經發行了有輕量版 Linux 之稱的 eCos 作業系統¹³⁴，但值得注意的是，截至目前為止，Red Hat 已經將 eCos 原始碼的著

¹³³ Red Hat Enterprise Linux 5.

¹³⁴ ecos Home Page, <http://ecos.sourceforge.org/> (last visited Mar. 9, 2011).

作權轉讓給 FSF¹³⁵。

在此同時，eCosCentric 也另外發行了功能較為強化的 eCosPro¹³⁶。簡單來說，eCosPro 係以 eCos 作業系統為基礎，提供了更多輔助功能與開發環境之軟體，而 eCosPro 必須透過商業授權取得。

eCos 的原始碼係以開放原始碼授權條款進行授權，被授權人可以在網路自由下載 eCos 的原始碼，但是 eCos 作業系統的授權方式中，有部份採用 GPLv2 授權條款而具有著佐權的效力¹³⁷；eCosPro 則是 eCosCentric 以商用授權的方式，提供廠商原始碼與技術服務，根據社群對 eCos 所回報的錯誤或修改來更新 eCosPro 的原始碼，進而使 eCosPro 的功能得以被調校，讓 eCosPro 具有較佳的效能。

易言之，儘管 Red Hat 同樣投注心力在作業系統的開發與維護，但是 Red Hat 對於個人電腦持續保持服務，但是對於嵌入式產品的作業系統則放棄其著作權。根據 Red Hat 的作法，亦可以推論出，因為個人電腦的技術服務市場較大，因此當 Red Hat 針對 Linux 作業系統提供技術資源時，客戶使用的數量較多，足以提供 Red Hat 生存的利基也較大，而不需要重新開發一套作業系統。但是就嵌入式產品而言，提供原始碼而促成的經濟效益有限，因此 Red Hat 最後在考量無法從中獲益的情況下，改為放棄原始碼的著作權而轉讓予 FSF。這也是針對個人電腦與嵌入式產品二者本質的差異，並進而衍生相對應的影響。

綜上所述，可以發現 Linux 作業系統與 eCos 作業系統雖然同屬於開放原始碼的作業系統，但其發展歷程卻不同，並連帶影響了其普及程度。Red Hat 最初採用與 Wind River 與 Monta Vista 不同的手段，而另外推出自己的一套作業系統，如此才得以享有雙重授權的方式，對於 eCos，Red Hat 可以享有全部的著作權，因此不必像 Wind River 與 Monta Vista 一樣受到 GPLv2 授權條款的限制，也不用以 Patch 的方式來分開散布原始碼，進而規避 GPLv2 授權條款的制約，而能夠完全避開 GPLv2 授權條款的掌控，對客戶也能提供完善的支援。然而，Red Hat 的授權策略也可能讓部分的開發者因為不想為商業軟體公司作嫁，而不願意投入 eCos 的開發，導致後續 eCos 無法蓬勃發展，進而使 Red Hat 放棄 eCos 的著作權。

在另一方面，這也顯示若非透過社群的力量來維護，對於單一廠商而言，開發、維護嵌入式產品的作業系統是一件相當吃力的事情。然而採用開放原始碼發展時，若採用了具有強烈著佐權效力之開放原始碼授權條款時，對於被授權人而言，拘束的程度又過

¹³⁵ eCos Copyright Assignment, <http://ecos.sourceware.org/assign.html> (last visited Mar. 9, 2011).

¹³⁶ eCosCentric, <http://www.ecoscentric.com/> (last visited Mar. 9, 2011).

¹³⁷ eCos License Overview, <http://ecos.sourceware.org/license-overview.html> (last visited Mar. 9, 2011).

大。因此，廠商在爲了降低發展成本而採用較爲便捷、現成的開放原始碼時，可以選擇由商用軟體服務公司所提供的版本，例如：Mona Vista、Wind River 對於 Linux 作業系統所提供的商業版本，以及 eCosCentric 所優化過的 eCosPro。儘管這些商用軟體服務公司所推出的版本仍有相當比重會受到 GPLv2 授權條款的影響，但是該些影響可以說是在使用開放原始碼軟體時，讓提供原始碼的影響降至最低的作法，也讓廠商在使用開放原始碼時，得到額外且特定的技術支援。

8.1.2.3 使用符合商業利益的開放原始碼授權條款所發行的軟體

如前所述，不同的開放原始碼授權條款賦予被授權人的義務並不同，因此廠商在使用開放原始碼時，應該要檢視所欲使用的開放原始碼，究竟採用何種開放原始碼授權條款，並了解各該開放原始碼授權條款所規範的內容。

單就使用原始碼的限制而言，MPL 與 BSD 類型的開放原始碼授權條款，對於被授權人必須提供原始碼的程度與強制性較低。假設被授權人所需要的功能，有不同的開發者提供原始碼，並以不同的開放原始碼授權條款進行授權，在開放原始碼有不同版本而具有選擇性的情況下，如果廠商對於原始碼的後續提供感到疑慮時，廠商便應選擇以揭露義務較低的 MPL 或 BSD 類型之開放原始碼授權條款進行授權的開放原始碼版本。

換句話說，廠商應該根據自身是否願意配合提供原始碼而決定原始碼來源的選用，例如，對於不願意提供原始碼的廠商而言，便應該避開以 GPL 類型之開放原始碼授權條款進行授權的原始碼，改爲選擇 Apache、BSD 與 MIT 等限制較少的開放原始碼授權條款的原始碼。對於不確定產品開發走向之廠商，本研究亦建議，在選擇使用開放原始碼時，盡量選擇拘束較少之開放原始碼授權條款，以免日後在受到必須提供原始碼之拘束時，導致廠商自己已經沒有他種選擇方式而被迫配合。

舉例來說，Apple 同樣是利用開放原始碼的資源，以 FreeBSD 作爲自家作業系統的基礎¹³⁸，對於 Apple 而言，不但可以利用開放原始碼的資源改善自家產品的缺失，也讓應用程式的開發者在原始碼可得的情況下，會較微軟的視窗作業系統更樂於以 Mac OS 作爲開發的平台，然而 Apple 因爲採用的是 BSD 授權條款的原始碼，並非採用 GPLv2 授權的原始碼，因此 Apple 不需要面臨將程式的原始碼全面提供給被授權人的問題¹³⁹。

¹³⁸ Introduction to Mac OS X Technology Overview, http://developer.apple.com/library/mac/#documentation/MacOSX/Conceptual/OSX_Technology_Overview/About/About.html (last visited Mar. 8, 2011).

¹³⁹ WEBER, supra note 61, at 201-202.

8.1.2.4 對程式開發過程進行相關防護

此外，嵌入式產品的開發廠商在開發系統程式時，針對開放原始碼的使用亦應相當注意，爲了確保自家產品所使用的系統程式符合開放原始碼授權條款的規定，嵌入式產品的開發廠商可以利用法務認知面與技術面兩種方式來防護。前者指的是根據使用階段的不同，將產品的開發分成採用前的評估作業，以及使用過程中的一致認識。後者指的則是，利用技術手段來防止未經確認之開放原始碼被使用於產品中，由於開放原始碼的使用成爲勢之所趨，而開放原始碼的授權條款也令許多人感到疑慮，因此相關的技術防護軟體與服務也應運而生，借此技術防護面來彌補個人認知程度與對授權條款的瞭解程度不一等缺失。

以法務認知面而言，除了在使用開放原始碼前，由公司的法務部門研究其授權內容，並與研發部門一同確立開放原始碼的使用規則外，並對研發人員提供教育訓練與訂定公司規範。藉由宣導或是在職訓練課程等方式，讓參與開發程式的系統工程師瞭解使用開放原始碼所帶來的利弊得失，並樹立相關的原始碼使用規範，例如：完全不使用、採用何種類型的開放原始碼授權條款之原始碼可被使用於產品中，或是需經過公司內部之特定成員的判斷後才使用其原始碼等措施，藉以確保所開發的系統程式符合開放原始碼授權條款的規定。

另一方面，技術防護面指的是：利用軟體檢驗程式來檢視公司在開發系統程式時，開放原始碼的使用是否合乎各種開放原始碼授權條款的限制，以自動化的軟體判斷程式原始碼是否使用了開放原始碼，進而避免人爲的疏漏。目前有的廠商推出程式原始碼的分析軟體¹⁴⁰，讓廠商得以分析自家產品所使用的原始碼是否之風險；有的廠商則推出原始碼開發的管理平台¹⁴¹，讓開發系統程式的嵌入式產品廠商便於管理不同來源的原始碼，與該等原始碼可使用之授權範圍等條件，用來確保廠商所開發出來的系統程式不會誤用未經合法授權的原始碼，或使用了被限制不得作爲商業使用的授權條款所開發的原始碼。除了商業公司外，目前也有一套軟體 FOSSology 係以開放原始碼的方式分享¹⁴²，這對我國的嵌入式產品開發廠商而言，不啻是個好消息。

¹⁴⁰ PALAMIDA, Application Security Products, <http://www.palamida.com/products> (last visited Apr. 15, 2010).

¹⁴¹ 如 Black Duck Software 所推出的 Black Duck Protex 平台。

¹⁴² 自由軟體授權分析輔助工具：自動化程式碼掃描系統：

<http://www.openfoundry.org/tw/legal-column-list/8094-2010-07-24-14-11-18>（最後點閱時間：2011年3月9日）。

8.1.2.5 配合提供相關原始碼

在 GPLv2 授權條款規範的內容無法修改的情況下，嵌入式產品的開發廠商仍可能迫於壓力而必須提供原始碼，在這種情況下，嵌入式產品的開發廠商仍可以利用一些手段來將提供原始碼的附帶效應降低。本研究對此提出的建議為：利用科技保護措施，以及延遲提供原始碼的時點。儘管這兩種方式並非自由軟體愛好者所樂見的行爲，但是這些手段確實符合 GPLv2 授權條款的文義規定，對於身為自由軟體被授權人的台灣廠商而言，本研究認為這兩種手段將成爲嵌入式產品的開發廠商偏好使用的方式。

事實上，自由軟體的開發社群認爲會選擇這些方式的廠商尚未體認到原始碼共享的真諦，所以才會有將原始碼的分享，視爲與助長競爭對手能力的想法，事實上，社群認爲自由軟體的開發模式已經顛覆了過去的產品開發思維，因爲廠商使用開放原始碼時，一般的使用者也可能對產品製造商提供解決問題的方案，讓廠商得以享受外來的資源，進而節省廠商的成本開銷¹⁴³。社群也認爲，台灣廠商這種與對手處處競爭的心態將無法維持太久，因爲在面對中國與印度的低成本競爭下，如果廠商無法開誠布公的分享系統程式的原始碼，將使產品的品質降低而無法進一步提供誘因吸引消費者選用。換句話說，廠商若採用科技保護措施或延遲提供的手段，將使自己的產品處於一個相對封閉的開發環境，而這樣的行爲就短期而言固然保障眼前的獲利，卻可能帶來未來的負面影響。

8.1.3 結論

開放原始碼授權條款可能衍生的議題相當繁複，但其重要性卻不容忽視，因此對於工程領域的系統軟體開發者與法律領域的律師或法務都有瞭解的必要。也因爲，實際應用開放原始碼的評估時，亦需要上述參與者的溝通與瞭解，相較於技術人員的參與，過去關於法律層面的關注仍明顯不足，因此需要更多的合作，才能使開放原始碼所採用的合作模式，得以有較佳的法律模型來支援¹⁴⁴。誠如前述，目前 Embedded Linux 的使用因爲有 Torvalds 的聲，讓廠商得以在 Embedded Linux 時稍微鬆綁，但是對於其他使用 GPLv2 授權條款的眾多開放原始碼而言，GPLv2 授權條款所衍生的法律風險，卻不因此而減少，因此廠商仍須密切注意。

在開放原始碼相當普及的現在，廠商在使用開放原始碼的時候，便不該抱持著與使

¹⁴³ 歐洲自由軟體基金會資深顧問 Shane Coughlan 於 2010 年 5 月 11 日，在自由軟體鑄造場所舉辦的活動（善用新一代的產業智慧：從自由軟體授權相容談起-自由軟體授權應用與商業建議二十講系列之三）所提及。

¹⁴⁴ See Karen F. Copenhaver, *Platform: Collaboration among counsel*, Vol. 1, Issue 1 International Free and Open Source Software L. REV., 53, 59(2009).

用專屬軟體時相同的封閉心態，也不應將原始碼認為是私有財產，而應該以開放的心態來確實瞭解開放原始碼授權條款的規定，特別是其中關於分享原始碼的要求與限制，並體認開放原始碼強調分享的價值。

儘管嵌入式產品廠商可能心存僥倖，認為自己所販售的產品並不會使用編譯前的原始碼，而編譯後的目的碼不但不容易理解，且被燒錄於晶片中，亦即，廠商使用開放原始碼的方式不易以肉眼判讀，也難以確認其原始碼的來源，在舉證不易的情況下，似乎可以避免使用開放原始碼的情形被發現。事實上，針對目的碼不易讀取的問題，目前開放原始碼的社群，已經發展出可以自動檢測二進位格式的目的碼中，是否含有開放原始碼的分析軟體¹⁴⁵。雖然該分析軟體所支援之判讀對象尚不多，但可以預見的是，該些檢測軟體將成為社群在舉證廠商使用開放原始碼時的利器。

如此一來，目的碼雖然已經經過編譯而不易由肉眼閱讀、判斷原始碼的形式與來源，但自動化檢測軟體的輔助下，對於目的碼在編譯前，是否使用開放原始碼的判斷難度已經降低，而廠商一旦違反開放原始碼授權條款規定時，其違反的行為也將越來越容易被發現。根據過去相關爭議的演變以及輔助工具的開發，對社群而言，可以用來與違反授權廠商相抗衡的力量亦相當足夠，也因此他們認為一旦提起訴訟，廠商必然是敗訴的一方，因此我國的嵌入式產品開發廠商應該更關切這方面的議題才是。

事實上，關於開放原始碼授權條款的選用，並沒有標準作業流程可以參照，甚至想要將需要參考的事項列成一個檢查表都不容易，以自由軟體鑄造場所提供的授權指引（精靈）(License Wizard)¹⁴⁶為例，目的雖然是要讓使用者可以依據所想要的特性加以選擇適合自己的授權條款¹⁴⁷，但是根據一般廠商的考量而選擇的話，現有的開放原始碼授權條款所能提供給廠商的選擇，很可能形成一個空集合。再者，在具有著佐權效力的開放原始碼授權條款中，廠商一旦使用了開放原始碼，便處於一個無從選擇後續授權方式的局勢，因此廠商對於開放原始碼授權條款更應該有充分的理解才是。

在「世上沒有白吃的午餐」的情況下，嵌入式產品的開發廠商固然可以利用極低的價格甚至以免費的方式取得程式原始碼，但也代表後續必須額外進行原始碼的維護與開發，以及開放原始碼授權條款所賦予的提供原始碼，或標示原始碼的來源等義務，因此嵌入式產品的開發廠商在使用開放原始碼時，應該將各方面的考量均納入權衡並進行取

¹⁴⁵ binary analysis, <http://www.binaryanalysis.org/en/home> (last visited Mar. 9, 2011).

¹⁴⁶ 自由軟體鑄造場, License Wizard 網站: <http://www.openfoundry.org/Law-and-Policy/License-Wizard/> (最後點閱時間: 2010年6月6日)。

¹⁴⁷ 需注意的是，由於開放原始碼授權條款的種類繁多，此處僅提供了十種較為常見的開放原始碼授權條款，且其中尚未包含在 2007 年釋出的 GPL 授權條款第三版 (GNU Public License Version 3, 簡稱為 GPLv3)。

捨¹⁴⁸，避免因為疏忽或無知而產生使用開放原始碼的後續問題。

8.2 未來研究建議

本研究所探討的 Embedded Linux 僅是開放原始碼授權條款在實務應用中，可能衍生的眾多法律議題中的一個例子，誠如前述，嵌入式產品的系統開發涉及了相當多的環節，且相關軟體的應用與分層等皆有可能涉及開放原始碼的使用。此外，嵌入式產品的開發廠商所使用的開放原始碼中，若包含了未被授權的軟體專利或未獲授權的其他軟體原始碼時又該如何？至於是否可能在未來有更多與開放原始碼授權條款相關的判決產生，進而讓開放原始碼授權條款的相關討論發展到另一個層次，也相當令人期待。

相較於已經推行許久的開放原始碼外，開放硬體與開放文件的推行也開始蓬勃的進行中，開放硬體指的是將硬體的設計細節加以分享，而開放元件則指文件的分享。相較於軟體，開放硬體的另外有其實際限制，因為硬體的開發需要相當高的成本，且硬體是否使用了開放原始碼的設計又較開放原始碼難追蹤¹⁴⁹，因此相關的搭便車行為也更難預防。

另一方面，即便是軟體開發者，對於他人所撰寫的原始碼，也經常感到難以閱讀與理解，若僅依據開放原始碼授權條款規定而單獨提供原始碼時，分享原始碼本身所能達成的效益相當有限，因為原始碼的架構與流程等，往往需要相關的說明文件來輔助，因此有些人主張應該同樣以 GPL 授權條款來對該些文件授權。GNU 對文件的部份亦提供了 GNU 自由文件許授權條款（GNU Free Documentation License，簡稱為 GFDL）¹⁵⁰，另外還有些人則主張應該使用針對文件所提供的創用 CC(Creative Commons)¹⁵¹來授權，這些問題也都與開放原始碼的推行息息相關，也是未來可以進行討論與研究的主題。

¹⁴⁸ Westinghouse 在 BusyBox 案件之侵權賠償解析-企業應規劃衡量利用自由軟體的法律風險：
<http://www.openfoundry.org/tw/legal-column-list/8144-westinghouse-busybox->（最後點閱時間：2011 年 3 月 9 日）。

¹⁴⁹硬體的產生尚需要進行合成、佈局與佈線（place and route）等步驟，再經過晶片製造的過程後，原始的硬體描述語言完全不復見，因此要判別是否使用相同的設計相當困難。

¹⁵⁰ GNU Operating System, GNU Free Documentation License, <http://www.gnu.org/licenses/fdl.html>, (last visited May 13,2010).

¹⁵¹ Creative Commons, <http://creativecommons.org/> (last visited Jun. 6, 2010).

參考文獻

中文書籍

Layder, Derker 著，簡守邦譯，《社會科學研究的理論與方法：從落實理論到調適理論》，(2002)。

Marshall, C.等著，李政賢譯，《質性研究：設計與計畫撰寫》，(2007)。

石之瑜，《社會科學方法新論》，(2003)。

邱聰智，《新訂民法債編總則(上)》，台北(2003)。

中文期刊

王凱，〈開放原始碼授權模式分析〉，《產業透析，軟體與應用透析》，第12期，頁2-10，2003年。

李科逸，〈自由軟體國際間智財權重要爭訟及趨勢之淺析-著作權訴訟爭議漸息，專利權侵權疑慮頻生〉，《智慧財產權月刊》，第76期，頁5-27，2005年。

Hsieh, Kuo-lien (謝國廉)，〈How Does the GNU General Public License Demonstrate Idealism in the Free Software Community?〉，《世新法學》，第1卷第1期，頁199-220，2007年。

陳人傑，〈開放原始碼授權條款之法律分析-以通用公共授權為中心(上)〉，第14卷第6期，《科技法律透析》，頁46-52，2002年。

陳人傑，〈開放原始碼授權條款之法律分析-以通用公共授權為中心(下)〉，第14卷第7期，《科技法律透析》，頁48-58，2002年。

馮震宇，〈開放原始碼授權契約之法律問題與挑戰〉，《全國律師》，第9卷第1期，頁3-17，2005年。

楊千、廖先志、陳鍾誠，〈從我國法制論 GPL 授權契約-兼論第3版草案之適用〉，《台灣科技法律與政策論叢》，第3卷第3期，頁121-150，2006年。

中文學位論文

李界昇，《開放原始碼模式下的法律糾紛與風險-從 SCO v. IBM 案出發》，交大科法所碩士論文，2004年7月。

李憲隆，《開放原始碼法律問題之研究-智慧財產權制度之檢討》，中大產經所碩士論文，2003年7月。

巫坤霖，《由著作權到著佐權-從著作權法及競爭法交錯之角度分析電腦程式著作授權契約》，台大法研所碩士論文，2009年7月。

林誠夏，《自由軟體產業推動之反思-以加拿大資訊產業發展政策為比較觀點》，台大國發所碩士論文，2007年6月。

林靜君，《論科技保護措施立法對著作權公共領域之衝擊》，逢甲財法所碩士論文，2008年7月。

康雲龍，《論開放原始碼軟體對智慧財產權理論之影響---以著作權法為中心》，銘傳法研所碩士論文，2007年7月。

陳儷卿，《開放原始碼軟體授權之相關法律風險分析》，交大科法所碩士論文，2010年7月。

其他中文參考文獻

Android 不是那麼開放，http://heaven.branda.to/~thinker/GinGin_CGI.py/show_id_doc/397 (最後點閱時間：2010年5月12日)。

Android 還是開放源碼計劃嗎？

<http://cwhuang.info/2010/01/is-android-still-an-open-source-project> (最後點閱時間：2010年5月12日)。

CLDP -Linux 中文文件計劃：<http://www.linux.org.tw/CLDP/OLD/doc/GPL.html> (最後點閱時間：2010/5/10)。

CNET News 網站：

http://news.cnet.com/Top-Linux-programmers-pan-GPL-3/2100-7344_3-6119372.html (最後點閱時間：2010年3月22日)。

GPL 的另類利用方式：「分開散布·責任轉嫁」：

http://www.openfoundry.org/component/option,com_content/Itemid,353/id,1711/lang,tw/task,view/ (最後點閱時間：2010年5月22日)。

Westinghouse 在 BusyBox 案件之侵權賠償解析-企業應規劃衡量利用自由軟體的法律風險：<http://www.openfoundry.org/tw/legal-column-list/8144-westinghouse-busybox-> (最後點閱時間：2011年3月9日)。

自由軟體授權分析輔助工具：自動化程式碼掃描系統：

<http://www.openfoundry.org/tw/legal-column-list/8094-2010-07-24-14-11-18> (最後點閱時間：2011年3月9日)。

自由軟體鑄造場，License Wizard 網站：

<http://www.openfoundry.org/Law-and-Policy/License-Wizard/> (最後點閱時間：2010年6月6日)。

從 BusyBox 案談起：台灣業者侵權利用自由軟體所面對的法律風險：

<http://www.openfoundry.org/news/2277> (最後點閱時間：2011年3月7日)。

著作權筆記，何謂數位權利管理：

<http://www.copyrightnote.org/crnote/bbs.php?board=3&act=read&id=530> (最後點閱時間：2010年5月5日)。

英文書籍

DIXON, ROD, OPEN SOURCE SOFTWARE LAW (2004).

GUIBAULT, LUCIE & DAALLEN, OT VAN, OPEN SOURCE IN PRACTICE, UNRAVELLING THE MYTH AROUND OPEN SOURCE LICENSES – AN ANALYSIS FROM A DUTCH AND EUROPEAN LAW PERSPECTIVE (2006).

LAURENT, ANDREW M. ST., UNDERSTANDING OPEN SOURCE AND FREE SOFTWARE LICENSING (2004).

LESSIG, LAWRENCE, FREE CULTURE-HOW BIG MEDIA USES TECHNOLOGY AND THE LAW TO LOCK DOWN CULTURE AND CONTROL CREATIVITY (2004).

ROSEN, LAWRENCE, OPEN SOURCE LICENSING – SOFTWARE FREEDOM AND INTELLECTUAL PROPERTY LAW (2005).

VÄLIMÄKI, MIKKO, THE RISE OF OPEN SOURCE LICENSING – A CHALLENGE TO THE USE OF INTELLECTUAL PROPERTY IN THE SOFTWARE INDUSTRY (2005).

WEBER, STEVEN, THE SUCCESS OF OPEN SOURCE (2004).

英文期刊

Benkler, Yochai, *From Consumers to Users: Shifting the Deeper Structures of Regulation toward Sustainable Commons and User Access*, 52, FEDERAL COMMUNICATIONS LAW JOURNAL 561 (2000).

Gonzalez, Andres Guadamuz, *Viral Contracts or Unenforceable Document? Contractual Validity of Copyleft Licenses*, 26(8), EUROPEAN INTELLECTUAL PROPERTY REVIEW 331 (2004).

Lessig, Lawrence, *The Architecture of Innovation*, 51 Duke Law Journal 1783 (2002).

Copenhaver, Karen F., *Platform: Collaboration among counsel*, Vol. 1 INTERNATIONAL FREE AND OPEN SOURCE SOFTWARE LAW REVIEW 53 (2009).

Mockus, Audris et., *Two Case Studies of Open Source Software Development: Apache and Mozilla*, 11 ACM TRANSACTIONS ON SOFTWARE ENGINEERING AND METHODOLOGY 309 (2002).

McGhee, Douglas D., *Free and Open Source Software Licenses: Benefits, Risks, and Steps*

Toward Ensuring Compliance, 19 INTELLECTUAL PROPERTY & TECHNOLOGY LAW JOURNAL 5 (2007).

Murray, Gwyn Firth, *Categorization of Open Source Licenses: More Than Just Semantics*, 26 THE COMPUTER & INTERNET LAWYER 1 (2009).

Pianon, Alberto, *Trade Secret Vs. Open Source: And The Winner Is ...*, 1 ERASMUS LAW AND ECONOMICS REVIEW 47 (2004).

Rosen, Lawrence, *Bad facts make good law: The Jacobsen case and Open Source*, 1 INTERNATIONAL FREE AND OPEN SOURCE SOFTWARE LAW REVIEW 27 (2009).

Rychlicki, Tomasz, *GPLv3: New Software Licence and New Axiology of Intellectual Property Law*, 30 EUROPEAN INTELLECTUAL PROPERTY REVIEW 232 (2008).

Walsh, Edmund J. & Tibbetts, Andrew J., *Reassessing the Benefits and Risks of Open Source Software*, 22 INTELLECTUAL PROPERTY & TECHNOLOGY LAW JOURNAL 9 (2010).

Wright, Bradley C., *Linux Users Risk Infringement*, 13 INTELLECTUAL PROPERTY & TECHNOLOGY LAW JOURNAL 1 (2001).

其他英文參考文獻

7th Circuit Holds GPL and Open Source Software Do Not Violate Antitrust Law, <http://www.techlawjournal.com/alert/2006/11/10.asp> (last visited Mar. 6, 2011).

A Legal Issues Primer for Open Source and Free Software Projects, See <http://www.softwarefreedom.org/resources/2008/foss-primer.html> (last visited Jun. 6, 2010).

A Practical Guide to GPL Compliance, <http://www.softwarefreedom.org/resources/2008/compliance-guide.pdf> (last visited Mar. 21, 2010) .

Android Developers, Terms and Conditions, <http://developer.android.com/sdk/terms.html> (last visited Apr. 11, 2010) .

Android Open Source Project, <http://source.android.com/license> (last visited Apr. 11, 2010).

Application Security Products, <http://www.palamida.com/products> (last visited Apr. 15, 2010).

BusyBox and the GPL Prevail Again,

<http://www.groklaw.net/article.php?story=20100803132055210> (last visited Feb. 7, 2011).

binary analysis, <http://www.binaryanalysis.org/en/home> (last visited Mar. 9, 2011).

BusyBox Developers Agree to End GPL Lawsuit against Verizon,

<http://www.softwarefreedom.org/news/2008/mar/17/busybox-verizon/> (last visited Mar. 6,

2011).

BusyBox Developers and High-Gain Antennas Agree to Dismiss GPL Lawsuit,
<http://www.softwarefreedom.org/news/2008/mar/06/busybox-hga/> (last visited Mar. 6, 2011).

BusyBox Developers and Xterasys Corporation Agree to Settle GPL Lawsuit,
<http://www.softwarefreedom.org/news/2007/dec/17/busybox-xterasys-settlement/> (last visited Mar. 6, 2011).

BusyBox GPL Software Notice, <http://us.zyxel.com/Support/GPL-OSS/BusyBoxNotice.aspx>
(last visited Mar 7, 2011).

Creative Commons, <http://creativecommons.org/> (last visited Jun. 6, 2010).

D-Link - GNU General Public License Disclosure, <http://tsd.dlink.com.tw/GPL.asp> (last visited May 21, 2010).

eCosCentric, <http://www.ecoscentric.com/> (last visited Mar. 9, 2011).

137 eCos License Overview, <http://ecos.sourceware.org/license-overview.html> (last visited Mar. 9, 2011).

eCos Copyright Assignment, <http://ecos.sourceware.org/assign.html> (last visited Mar. 9, 2011).

ecos Home Page, <http://ecos.sourceware.org/> (last visited Mar. 9, 2011).

Estimating the Total Development Cost of a Linux Distribution
<http://www.linuxfoundation.org/publications/estimatinglinux.php> (last visited Jun. 6, 2010).

First U.S. GPL lawsuit heads for quick settlement,
<http://www.linux-watch.com/news/NS3761924232.html> (last visited Apr. 13, 2010).

Fortinet (UK) Ltd. Reaches Amicable Settlement Agreement with gpl-violations.org Project,
http://www.fortinet.com/press_releases/050425_gpl.html (last visited May 21, 2010).

Free Software Comes at a Price: Violation of Open Source License Leads to Enhanced Monetary Damages and Forfeiture of Products in Software Freedom Conservancy v. Best Buy et al., <http://www.fr.com/free-software/> (last visited Feb 7, 2011).

GNU Free Documentation License, <http://www.gnu.org/licenses/fdl.html>, (last visited May 13, 2010).

GNU General Public License, <http://www.gnu.org/licenses/gpl.html> (last visited Mar. 9, 2011) .

GNU General Public License, version 2,
<http://www.gnu.org/licenses/old-licenses/gpl-2.0.html> (last visited May 10, 2010) .

GNU Operating System, <http://www.gnu.org/licenses/gpl-2.0.html> (last visited Mar. 22,

2010) .

GPL violation, <http://gpl-violations.org/> (last visited Apr. 13, 2010).

Gpl-violations.org, gpl-violations.org project was granted a preliminary injunction against Fortinet UK Ltd., <http://gpl-violations.org/news/20050414-fortinet-injunction.html> (last visited May 21, 2010).

GPL Code Center, <http://homesupport.cisco.com/en-us/gplcodecenter>, (last visited May 22, 2010).

Harald Welte, http://en.wikipedia.org/wiki/Harald_Welte (last visited Dec. 27, 2009).

Harald Welte' s blog, <http://laforge.gnumonks.org/weblog/index.html> (last visited Dec. 27, 2009) .

International free and open source software law review, <http://www.ifosslr.org/ifosslr> (last visited May 13, 2010).

Introduction to Mac OS X Technology Overview,

http://developer.apple.com/library/mac/#documentation/MacOSX/Conceptual/OSX_Technology_Overview/About/About.html (last visited Mar. 8, 2011).

JBB Rechtsanwälte, http://www.jbb.de/judgment_dc_frankfurt_gpl.pdf (last visited Mar. 7, 2011).

Openmoko™ - Open. Mobile. Free http://wiki.openmoko.org/wiki/Main_Page, (last visited May 22, 2010).

Licenses, <http://source.android.com/source/licenses.html> (last visited Mar. 8, 2011).

Licensing Policy, <http://meego.com/about/licensing-policy> (last visited Mar. 8, 2011).

Licenses by Name, <http://www.opensource.org/licenses/alphabetical> (last visited, Mar 7, 2011).

Open Source Licenses, <http://www.microsoft.com/opensource/licenses.mspx> (last visited Jan. 17, 2010).

OPEN SOURCE LICENSE SERVICES, http://brainleague.com/offering/open_source.htm (last visited Jun. 5, 2010).

OSRM Position Paper: Mitigating Linux Patent Risk,

http://www.osriskmanagement.com/pdf_articles/linuxpatentpaper.pdf (last visited, Mar. 9, 2011).

Overview: the TRIPS Agreement, http://www.wto.org/english/tratop_e/trips_e/intel2_e.htm, (last visited Jun. 4, 2010).

Products/Projects Using BusyBox, <http://www.busybox.net/products.html>, (last visited May

22, 2010).

Skype Will be Open Sourced, <http://linuxcrunch.com/content/skype-will-be-open-source>
(last visited Dec. 27, 2009) .

Richard Stallman's Personal Home Page, <http://www.stallman.org/>, (last visited Feb. 20, 2011).

SFLC, <http://www.softwarefreedom.org/news/2009/dec/14/busybox-gpl-lawsuit/> (last visited Apr. 11, 2010) .

The Apache Software Foundation, Applying the Apache License, Version 2.0,
<http://www.apache.org/dev/apply-license.html> (last visited Apr. 11, 2010).

The Apache Software Foundation, Apache License v2.0 and GPL Compatibility,
<http://www.apache.org/licenses/GPL-compatibility.html> (last visited Apr. 11, 2010).

The Free Software Definition, <http://www.gnu.org/philosophy/free-sw.html> (last visited May 22, 2010).

The Linux Devices Showcase,
[http://www.linuxfordevices.com/c/a/Linux-For-Devices-Articles/The-Linux-Devices-Showcas](http://www.linuxfordevices.com/c/a/Linux-For-Devices-Articles/The-Linux-Devices-Showcase/)
e/, (last visited Jan. 17, 2010).

The Linux Edge , <http://oreilly.com/catalog/opensources/book/linus.html> (last visited Mar. 22, 2010).

Victory: Skype Withdraws Appeals Case, Judgement from Lower Court Accepted,
http://laforge.gnumonks.org/weblog/2008/05/08/#20080508-olg_muenchen-skype (last visited Mar. 6, 2011).

Where Embedded Open Source Goes Business-Critical,
<http://www.windriver.com/products/linux/>, (last visited May 22, 2010).

Why Upgrade to GPLv3, [http:// www.gnu.org/licenses/rms-why-gplv3.html](http://www.gnu.org/licenses/rms-why-gplv3.html) (last visited June 5, 2010).

附錄一 訪談資料

深度訪談說明書

您好：

我是交通大學科技法律研究所在職專班科技組的學生，基於碩士資格要求，本人在王敏銓教授與林三元法官的指導下，正著手畢業論文的研究。個人的研究主題在於開放原始程式之相關授權條款對於嵌入式系統開發時的法律問題，本研究結果將被使用於研究者本人的碩士學位論文，並可能發表於會議/期刊論文，希望您能分享個人的想法和經驗以協助本研究進行。

本研究的研究方法採用半結構式的深度訪談方式，約需 1~2 次的訪談，每次約需花費 30~60 分鐘，同時，基於資料整理與分析的考量，希望您同意讓我在訪談過程中錄音並整理筆記，錄音之內容僅供本研究之分析資料、編碼使用，基於保護受訪者的緣故，您的姓名與任職單位與工作內容等資訊在論文中將不公開，而以匿名及代碼呈現，因此，希望您能安心接受訪談以協助本研究的進行。

訪談期間您有權利選擇退出，且沒有義務告知原因，研究者本人會將錄音資料歸還給您，若您對本研究有任何意見，歡迎隨時提供給研究者本人，並誠摯的邀請您參與本研究。

十分感激您撥冗接受訪問，打擾之處，尚祈見諒！

國立交通大學科技法律研究所
論文指導教授：王敏銓 教授& 林三元 法官
在職專班科技組研究生：翁儷娟

中華民國九十八年十二月

深度訪談同意書

本人同意參與國立交通大學科技法律研究所在職專班研究生翁儷娟之碩士論文之訪談，就開放原始程式之相關授權條款對於嵌入式系統開發時的法律問題分享個人經驗與意見，並同意研究者於訪談過程全程錄音，以利後續資料整理與分析工作之執行，惟訪談內容僅供此研究主題與研究者本人之使用，並須於整理後經匿名處理始得發表。

立同意書人：_____ (簽名)

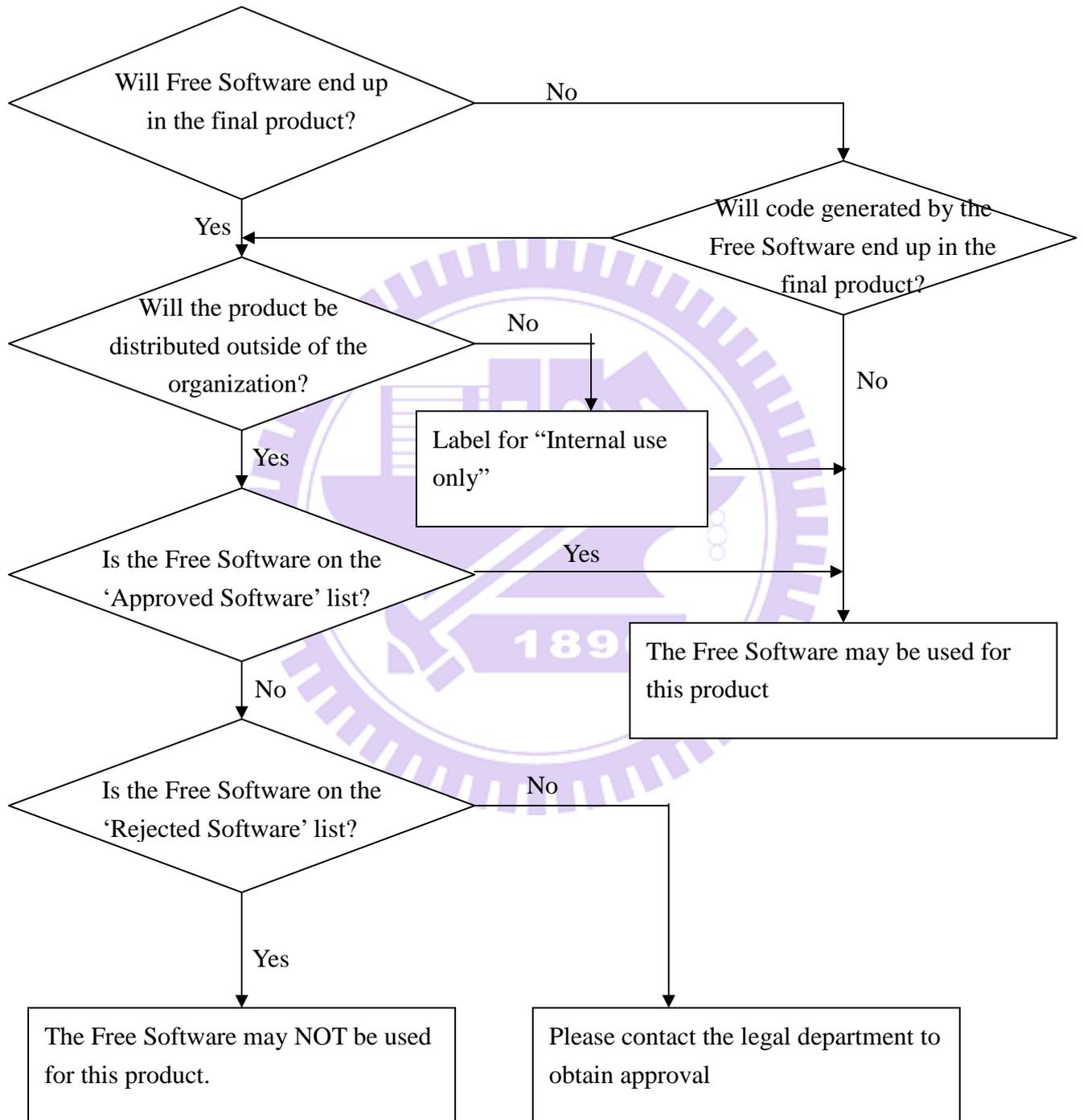
中華民國____年____月____日

訪談大綱

- 系統程式開發的作法與授權方式
 1. 以何種方式開發系統程式?
 2. 開發系統程式時的各層程式與開發環境來源?
 3. 系統程式開發時的管理方式為何?
 4. 系統中是否使用作業系統?程式來源?
 5. 是否曾經提供原始碼給客戶/使用者?主動/被動提供?
 6. 在什麼情況下會主動/被動提供原始碼?
 7. 在什麼情況下曾被要求提供原始碼?
 8. 被要求提供程式時會打算以何種方式開放?
- 自由軟體在系統程式開發的使用
 1. 曾經使用網路或其他管道搜尋/取得程式?
 2. 對於對自由軟體的瞭解? (who, where, why, how, when)
 3. 對於自由軟體的看法? 評價?

4. 認為自由軟體對於系統開發時會不會有幫助?什麼樣的幫助?
 5. 是否曾經以個人身分/以公司角色參與過自由軟體的開發?
- 軟體開發時與軟體授權相關的法律議題(排除軟體專利, 僅考慮軟體授權的問題)
1. 對於專業技術可能涉及的法律問題, 貴公司是否對於研發人員提供相關課程?
 2. 過去的經驗曾否接受相關的法律課程/訓練?
 3. 是否想過程式的開發過程會引發法律問題?
 4. 認為什麼樣的原因會使程式的開發與法律產生關聯?
 5. 若系統開發時被要求公開/提供程式碼時會有哪些考量?
 6. 除了上述問題外, 對於軟體開發時的相關法律問題有無其他的看法或意見?
- 使用自由軟體時的法律爭議
1. 不同類型(BSD, MPL, GPL)的拘束性與否對於程式開發的管理/使用造成問題?
 2. 是否關心過相關的訴訟, 是否知道判決結果?
 3. (告知相關的判決結果後) 認為這樣的判決是否合理?法律上?實際上?
 4. 若被自由軟體的所有權人宣稱所開發的程式被使用於貴公司的系統程式, 會以什麼樣的方式來處理?
 5. 除了上述問題外, 對於系統開發若使用了開放原始碼開發還有什麼看法?
- 最後, 在結束訪談前, 對於上述訪談過程, 是否有可以補充/建議之處?

附錄一 自由軟體使用判斷流程圖¹⁵²



¹⁵² Royal Philips Electronics 對於是否應用開放原始碼在產品中的判斷流程圖。

附錄三 GPLv2 授權條款

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the

danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License.

(Exception: if the Program itself is interactive but does not normally print such an

announcement, your work based on the Program is not required to print an announcement.) These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a)** Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b)** Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c)** Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy

the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY

OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS



簡歷

經歷：

博拓智權事務所 專利一部 (2010.6.~今)

專利工程師

碩維智權事務所 (2009.7.~2010.5.)

專利工程師

聯詠科技股份有限公司 光儲存多媒體產品事業部 (2008.7.~2009.6.)

系統應用工程師

其樂達科技股份有限公司 數位廣播系統開發處 (2006.3.~2008.6)

系統開發工程師

工業技術研究院 系統晶片技術發展中心 (2003.10.~2006.3.)

電子設計自動化工程師

學歷：

國立交通大學 管理學院 科技法律學程 (2006.7.~今)

碩士在職專班

愛爾蘭都柏林市立大學 電子工程研究所 (2001.10.~2004.10.)

碩士

國立交通大學 電機與控制工程系 (1997.9.~2001.6.)

學士