Free Software Licensing,

a Draft White Paper by CONNOTECH Experts-conseils inc.

Document Number C002226

2003/11/29

Document Revision History

C-Number	Date	Explanation
C002226	2003/11/29	First Release
C002226		Current version

1. Introduction

The ABCD Proto-Kernel™ software uses GPL-style licensing scheme. The specific licensing terms allow, under certain circumstances, to distribute proprietary applications based on the ABCD Proto-Kernel™. This approach reconciles the free-software business model with the practicalities of ent embedded market, where software is implemented in *devices* rather than on a computer system with a clean separation between the operating system and software applications. This white paper attempts to guide the reader through the meanders of GNU licensing in general, and does so by following the same mental process followed to devise the license terms used by CONNOTECH in its various free software distributions.

Free Software Foundation's GPL licensing is the foundation of significant software development initiatives. As a legal framework with obligations carried to users and contributors of free software enhancements, its details can not be totally ignored. The GPL licensing scheme is based on the copyright law and unique additional provisions carefully crafted to create a self-fueling free software development momentum. The wording of the GPL licenses is such that many software engineers are likely to understand the terms and figure out how they apply in a given situation.

In this document section, you will see how the GPL licensing scheme is applied to the ABCD Proto-Kernel™ in an attempt to participate in the free software development momentum *and* to allow, under certain circumstances, proprietary processes implemented as proprietary embedded software applications based on the ABCD Proto-Kernel™.

2. Definitions

Software bundle

> A software bundle is a software program, distributed in any format or media, including notably an embedded software that is part of a device, system, or process involving digital processors. In the GPL terminology, a software bundle can be “a whole which is a work based on the Program” when the software bundle is to be licensed with the ordinary GPL, or otherwise a “work that uses the Library” when the software bundle contains some software components licensed with the LGPL.

Software component

> A software component usually comes in source code form and is compiled and linked to become part of a software bundle. Logical grouping of source code files is recognized in the GPL wording “if identifiable sections of [a software bundle] are not derived from the Program [or Library], and can be reasonably considered independent and separate works in themselves.”

Programmatic interface

A free program can interact with a non-free program. For instance, a non-free program can run on a free operating system and vice-versa. So the licensing requirements usually do not cross programmatic interfaces between programs, materialized as output data formats, protocol specifications, and the like (but not application programming interfaces). The exceptions to this rule are explained in the GPL or LGPL text.

3. A Decision-Making Flowchart

The flowchart in figure 1 on page 5 may be followed to come up with a licensing strategy in actual situation. It represents the decision-making that occurred for the ABCD Proto-Kernel™ licensing using the LGPL with a waiver on the opportunity to relink.

Here are the comments about the steps and decision points in the flowchart:

1 Identify programmatic interfaces

This step allows you to determine what is the software bundle for which a licensing strategy is to be established. In doing so, you narrow he scope of licensing strategy decisions.

2 Preconceived opinion?

You or your organization may already have an opinion on the issue of free software development. It can be wise for a software developer to warn the managers that some source code may eventually be released to customers before the lawyers get nervous with this. The bulk of the flowchart applies when there is room for adapting to the specific situation of a given software development effort.

3 GPL?

Look at the table 1 on page 6: the areas labeled "A" and "B" represent the antagonism between FSF GPL-style licensing and other licensing approaches. That is, you can't have in the same bundle GPL'ed components and proprietary components (or GPL-incompatible or components that you write but decline to release in source form to your customers).

4 Follow FSF guidelines

If you take the orthodox GPL'ed software bundle approach, you have unrestricted access to a huge source code base from where you can retrieve any interesting piece. You put together source code components that are GPL-compatible. You may use guidance provided by the Free Software Foundation in this respect.

5 Proprietary or BSD-style licensing

If your organization can't tolerate *any* form of software source code release to customers, then you are in the wrong camp as far as this flowchart is concerned. If you feel unfair that you can't use some good quality GPL'ed software components, you might revise your opinion at some point.

If you insist on BSD-style licensing, the flowchart is also not applicable to you. That is because BSD-style licensing does not impose the very licensing restrictions that propel the free software momentum with GPL-style licensing.

6 Identify SW components

The software components of the software bundle to be developed should be identified with their respective licensing terms. Keep an eye on the GPL-incompatible software components that are introduced with little justification.

7 Any GPL-incompatible license?

The GPL-incompatible licenses include software components that you are not allowed to redistribute in source code form, including the original software that your organization developed internally and wants to keep secret. The Free Software Foundation maintains a list of prevailing licenses with indications as whether they are GPL-compatible or not.

GPL-incompatibilities are sometimes circumvented with “special exceptions” to the ordinary GPL license. Only the original contributor of the software component can make the decision to affix a special exception. The flowchart cannot accurately reflect the impact of such special exceptions, but they often behave as if licensed with the LGPL or the LGPL with a waiver on the opportunity to relink.

8 Embedded SW project?

With typical embedded software development projects, the kernel and application software components are linked together in a single software image file. The embedded software application interacts with the real world using sensors, actuators, and other direct connections with electronically controlled processes.

9 Out of scope

The present licensing strategy flowchart does not address software development projects outside of embedded software. Perhaps issues such as dynamic link libraries invalidate some of the decision flow elements. More significantly, the present flowchart is not intended to promote non-free software outside of the case where a proprietary process is implemented as an embedded software application.

10 Any GPL'ed component?

A GPL'ed software component is one which uses the ordinary GPL terms, without any special exception that applies to your situation.

11 Ordinary GPL licensing

The ordinary GPL licensing strategy is the one that best protect 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. You can use the ABCD Proto-Kernel™ with the ordinary GPL licensing strategy.

12 Future developments with GPL incompatibilities?

The licensing strategy at one point in time may anticipate future developments where GPL-incompatibilities are a possibility.

13 Any GPL'ed component?

A GPL'ed software component is one which uses the ordinary GPL terms, without any special exception that applies to your situation.

14 Remove a conflicting SW component

This step is where the significant software design decisions must be made. Generally, we advocate for the replacement of proprietary (or otherwise GPL-incompatible) components with free ones. But you may also whish to replace a GPL'ed component with a GPL-compatible one having lesser licensing requirements.

15 Reluctant to offer the opportunity to relink?

The opportunity to relink is worded in the LGPL in a way that permits “modification of the work for the customer's own use and reverse engineering for debugging such modifications.” When a proprietary process is implemented with an embedded software application, those terms might be unacceptable e.g. for safety or operational reasons, or for enforcement of the trade secret protection for an industrial process.

16 LGPL licensing

With the LGPL licensing strategy, you accept that distributing the software bundle forces you to offer the source code for the (LGPL-covered) libraries to your customers, and you accept that they can relink the software bundle with a modified version of the library. This is a good way to promote the use and development of free software. You can use the ABCD Proto-Kernel™ with the LGPL licensing strategy.

17 Any SW component with relink clause?

The ordinary LGPL states that “If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it.” This is the opportunity to relink. The opportunity to relink can also be present in a “GPL with a special exception” license based on the exact wording and your situation.

18 LGPL with waiver on opportunity to relink

With the “LGPL with a waiver on the opportunity to relink” as a licensing strategy, you accept that distributing the software bundle forces you to offer the source code for the (LGPL-covered) libraries to your customers, thus promoting the use and development of free software. Based on a carefully written copyright license (the LGPL), this licensing strategy nonetheless protects the proprietary nature of processes implemented with embedded software applications. You can use the ABCD Proto-Kernel™ with this licensing strategy.

4. Conclusion

The fairly detailed decision-making process explained in this document should give confidence to the reader that a proprietary software development may be undertaken with the ABCD Proto-Kernel™ as the microprocessor software foundation. The benefits of free software use, and the corresponding re-distribution obligations, represents a unique software engineering opportunity, provided that close attention is paid to the specific licensing terms of each software component put in a given software project.

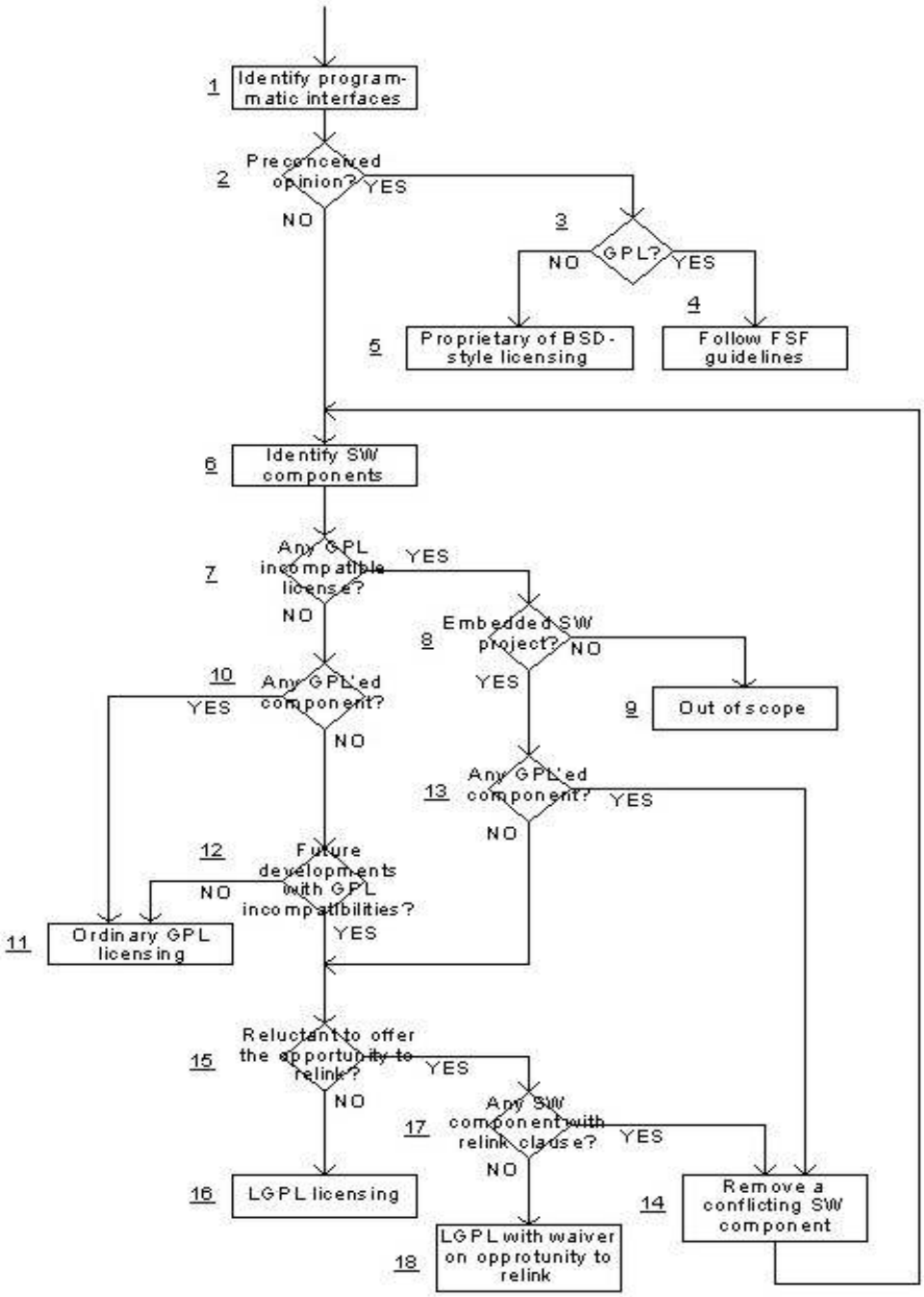

Figure 1) Flowchart for Embedded Software Development Licensing Strategy

<table>
<tr><th>Software bundle
Source components</th><th>Proprietary or GPL-incompatible</th><th>BSD-licen-sed or public domain</th><th>Embedded proprietary with free library source code</th><th>Proprietary with free library re-link opportunity</th><th>GPL'ed</th></tr>
<tr><td>Proprietary or GPL-incompatible</td><td>✔</td><td>✔</td><td>✔</td><td>✔</td><td rowspan="2">"A"</td></tr>
<tr><td>Original software not released</td><td>✔</td><td>✔</td><td>✔</td><td>✔</td></tr>
<tr><td>BSD-licen-sed or public domain</td><td>✔</td><td>✔</td><td>✔</td><td>✔</td><td>✔</td></tr>
<tr><td>(L)GPL'ed with one of the embedded exceptions</td><td colspan="2" rowspan="3">"B"</td><td>✔</td><td>✔</td><td>✔</td></tr>
<tr><td>LGPL'ed</td><td></td><td>✔</td><td>✔</td></tr>
<tr><td>GPL'ed</td><td></td><td></td><td>✔</td></tr>
</table>

Table 1) Software component vs bundle licensing matrix

Software bundle	Proprietary or GPL-incompatible	BSD-licen-sed or public domain	Embedded proprietary with free library source code	Proprietary with free library re-link opportunity	GPL'ed
Customer can rebuild the software bundle?	?	?	Not necessarily	Yes, application available in object form by license agreement	Yes
Customer has access to source code?	?	?	Partial	Partial	Yes, complete

Table 2) Software bundle properties

Source components	Source-code availability for paid distribution customers	Collective support incentive	Major functionality enhancements contribution incentive
Proprietary or GPL-incompatible	?	No	No
Original software not released	?	No	No
BSD-licensed or public domain	Uncertain	Limited	Limited
(L)GPL'ed with one of the embedded exceptions	Yes, license provision	Yes	Limited
LGPL'ed	Yes, license provision	Yes	Limited
GPL'ed	Yes, license provision	Yes	Yes

Table 3) Software component properties