# Driven to Tears – GPLv3 and the Automotive Industry

Jeremiah C. Foster,<sup>a</sup> (a) GENIVI Community Manager and FOSS enthusiast

DOI: <u>10.5033/ifosslr.v7i1.102</u>

#### Abstract

The automotive industry is moving toward the use of Free and Open Source software (FOSS) in vehicles. GPLv3 is currently presenting a roadblock to greater adoption. Specifically the Installation Information requirement in GPLv3 Section 6 (sometimes called the "Anti-Tivoization" clause) is causing some car makers to fear GPLv3. These car-makers want to lock down all software installed on their cars against user modifications, but fear that using GPLv3 software will prevent them from doing so. Although there may be good reasons to lock down some software on cars, car-makers should not fear GPLv3. One solution the industry may wish to consider to allay concerns about the Installation Information requirement in GPLv3 is to adopt and advocate for use of an "Additional Permission" that excepts users from having to comply with that requirement.

#### Keywords

GPLv3; Installation Information; Anti-Tivoization; automotive;

# Car makers and GPLv3: Current Concerns

In the last five years, the automotive industry has begun widely using Free Software.<sup>1</sup> Primarily used for handling media and providing services – such as navigation – FOSS has nonetheless made inroads into an industry that has historically relied on closed-source proprietary software. This cautious movement to Free and Open Source Software ("FOSS") has followed a predictable trajectory not unlike other industries which have discovered GNU/Linux and other FOSS software.<sup>2</sup> The embrace of FOSS software in the automotive industry, in particular software licensed under the GNU General Public License ("GPL"), has

International Free and Open Source Software Law Review

<sup>&</sup>lt;sup>1</sup> E.g., <u>http://projects.genivi.org/what</u>

<sup>&</sup>lt;sup>2</sup> E.g., <u>http://www.comparebusinessproducts.com/fyi/50-places-linux-running-you-might-not-expect</u>

lead to a certain amount of cost savings and improved quality. However, this embrace has not included GPLv3 – and specifically the Anti-Tivoization clause in that license – and the rejection of GPLv3 has been vehement enough to result in "blacklisting".<sup>3</sup> This blacklisting is considered necessary by those who advocate for it in order to prevent users from modifying the software on their vehicle, which is generally prevented by the locking of software onto hardware using cryptographic keys.

Locking the software to the hardware – by signing the original software image with a cryptographic key so that only an image provided by the supplier will boot or install – is a common practice in embedded devices.<sup>4</sup> This process of signing software images – so only images with the right key will boot or install – effectively prevents a user from modifying the software on the device since they have no access to the key needed to allow their modified version to boot or install. This practice was considered by the author of the GPL – Richard Stallman – to violate the spirit of the GPL, and resulted in the addition of the "Installation Information" obligation in GPLv3.<sup>5</sup>

Car makers want the ability to Tivoize the software on their vehicles to ensure that the user does not modify the software on the vehicle's head unit. The major reason claimed by car makers for locking the software on their vehicles is safety.

## **ECU Remapping and Software Locking**

The claim that complying with GPLv3 to allow a user to modify the software in a vehicle based on safety concerns is disingenuous. Drivers have, for many years, replaced parts of their car, such as tires, brakes or sometimes even software.<sup>6</sup> In addition, drivers frequently use off-brand or non-original parts, often because they're considerably cheaper but just as safe and functional. There is even a large after-market for remapping Engine Control Units ("ECUs").<sup>7</sup> ECUs are microprocessors which control fuel mixture, turbo charging, transmission, and other drive train features of the car, almost all of which in some way affect safety and performance. This after-market sells services like ECU remapping to increase performance or to improve fuel economy.<sup>8</sup> While the ECU remapping business is something of a grey market – since it is not fully supported by car makers and can increase the cost of your insurance and void a car's warranty<sup>9</sup> – nonetheless car makers are tacitly supporting this market. Car makers support ECU remapping by making companies that provide that service part of their motor sports stable of advisers, by using data from the ECU re-mappers to understand performance changes resulting from remapping, and generally looking the other way if customers decide to install re-mapped ECUs.<sup>10</sup> Even car dealers may have a hard time

<sup>&</sup>lt;sup>3</sup> See "LFCS: GPLv3 and automobiles" <u>https://lwn.net/Articles/548212/</u>

<sup>&</sup>lt;sup>4</sup> See "It's not just TiVo locking down their hardware" <u>https://www.fsf.org/blogs/licensing/gplv3-lockdown</u>

<sup>&</sup>lt;sup>5</sup> See "Transcript of Richard Stallman on GPLv3 in Brussels, Belgium; 1st of April 2007" http://fsfe.org/campaigns/gplv3/brussels-rms-transcript#tivoisation

<sup>&</sup>lt;sup>6</sup> See "WTF! It Should Not Be Illegal to Hack Your Own Car's Computer" <u>http://www.wired.com/2015/01/let-us-hack-our-cars/</u>

<sup>&</sup>lt;sup>7</sup> See "What is Remapping, and is it Worthwhile?" <u>http://www.moneysupermarket.com/car-insurance/blog/what-is-remapping-and-is-it-worthwhile/</u>

<sup>&</sup>lt;sup>8</sup> *Ibid.* 

<sup>9</sup> Ibid.

<sup>&</sup>lt;sup>10</sup> See "Ford Racing really wants you to mod EcoBoost engines" http://www.roadandtrack.com/car-culture/buying-maintenance/news/a24602/ford-racing-really-

spotting a non-original ECU and would therefore likely not refuse warranty service on an ECU re-mapped vehicle.<sup>11</sup>

Remapping an ECU can be dangerous. Changing the fuel mixture may not cause safety issues, but if you were to significantly increase the power of a car without commensurate changes in handling characteristics you might increase the risk of an accident. Safety issues certainly need to be considered when remapping an ECU. For these reasons, one would expect a similar reaction from the car manufacturers to ECU remapping as the current position on modifications to head unit software; namely, that it is forbidden for safety reasons and technological measures like cryptographic keys would be used to prevent it. That this is not widely the case raises the suspicion that there may be other reasons – other that safety – motivating some car manufacturers to prevent user-modifiable software in the head unit of their cars.

# Software: A New Revenue Driver for Car Manufacturers?

Speculating on those reasons is not hard to do. Car makers are becoming software producers and they are using this new capacity to market modern cars to appeal to contemporary drivers. Software is an opportunity not just to increase safety and performance but to engage the driver and passengers in a way that builds a relationship. Each update is an opportunity to strengthen that relationship, each point where the driver or passenger engages the software is an opportunity for the car makers to build that relationship further, and that relationship can represent an opportunity for significant revenues. These revenues would not necessarily be significant if they are just gathered through sales via a bespoke app store; the revenues from such a bespoke app store may be too low – and the costs of alteration of the relationship between the car vendor and the driver or passenger could be too high – to justify allowing modified software or applications.

What car makers likely want is a way to market new vehicles to younger drivers and to provide seamless and easy to use services to their middle-age customers, as well as to integrate modern notions of mobility and connectivity into their vehicles to appeal to a broad range of customers. Software is a key part of that marketing strategy. In fact, advertising tomorrow's technology manages to sell cars today. This is why we see so much press on the Apple and Google entrance into the In-Vehicle Infotainment ("IVI") market; the anticipation of these companies being connected with systems in a vehicle sells cars now even though it likely won't be widely seen in cars for years.

Preventing a user from changing the software in their car is likely driven by the desire to keep the in-car experience branded. The consequences of diluting that brand, either by blocking branded content, or by causing branded content to work in ways different than the brand owner desires, could result in loss of revenue through diminished brand loyalty, lost accessory sales, and even lost advertising – a business some car companies have stated they'll go into. There is likely a rich trove of data waiting to mined in the vehicle that car makers and others are eager to get a hold of, so as to target advertising. Keeping control over the In-Vehicle Infotainment system, the system that provides media, navigation, and connectivity and runs on the "head unit," is desirable. There is likely an incentive for car makers to try to mitigate the effects any license – like GPLv3 – which facilitates a user's modification of

wants-you-to-mod-ecoboost-engines/

<sup>&</sup>lt;sup>11</sup> *Ibid*.

International Free and Open Source Software Law Review

software on the head unit in a way that could impede data collection or advertisement targeting.

## Safety: Is It An Issue?

There is, however, some merit to the view that the car makers are not dressing up a commercial need under the guise of a safety-critical concern. Those who stand in the second rank on legal issues – right after the automotive legal team – state that with regard to the GPLv3, the difficulty is with only the Anti-Tivoization clause, and the reason for disfavoring that license is safety. That proposition is worth taking at face value if only to test some of the assumptions made.

Modern cars have around 100 million lines of code running on them, <sup>12</sup> with 70% of that code being in the head unit. Complexity is a non-trivial issue in automotive software design. In addition to being complex, cars can be dangerous. The World Health Organization says that:

[R]oad traffic injuries are the eighth leading cause of death, and as such are an important global public health problem. They are the number one cause of death among those aged 15-29 years. There were approximately 1.24 million road traffic deaths in the world in 2010, 77% of which were among males. Middle-income countries had the highest burden and the highest road traffic death rates. <sup>13</sup>

In the United States deaths in motor vehicles are a serious problem. While the U.S. has reduced deaths by drunk driving over the last few decades via public health advertising, ignition locks, and sobriety checkpoints, deaths are still very high in comparison to other countries. Regulation has a role to play in reducing automobile deaths, and that regulation will directly affect car makers – both how they construct cars and how they are liable for malfunctions.

Regulation in the auto industry is not typically a consideration for many FOSS developers. The GPL and other open source licenses typically disclaim any liability, so when using FOSS, automotive companies may not have the expectation that their suppliers will assume liability for harms resulting from their software. Either the car manufacturers will need to become comfortable that they must assume any liability for the FOSS that they use, or they will have to educate and change the culture of the FOSS software development houses that they hope to work with so as to reduce the potential for the car manufacturers having to take on substantial liability for the use of FOSS.

If an automotive company has to go to court, it often requires its software suppliers, via a contractual indemnity, to shoulder some or all of the legal burden resulting from that software. This would not occur when one uses software that disclaims any liability. In addition, because a global car company is selling into (or having its products operate in) myriad jurisdictions with myriad different rules for liability for products, ensuring safety of those products so as to reduce the manufacturer's liability costs can be complex and costly.

<sup>&</sup>lt;sup>12</sup> See "Codebases: Millions of Lines of Code" <u>http://www.informationisbeautiful.net/visualizations/million-lines-of-code/</u> and "Genivi: BMW Case Study" <u>http://www.genivi.org/sites/default/files/BMW</u> Case Study Download 040914.pdf

http://www.genty.org/sites/default/mes/bitw\_Case\_study\_Download\_04091
http://www.who.int/gho/road\_safety/mortality/number\_text/en/

Automotive software has a role to play in the liability equation, both in the way in which it may affect the driver and the vehicle. Whether it is measuring the cognitive workload on the driver, or assisted driving through monitoring the car ahead, software will be able to greatly assist drivers to drive more safely. Not preventing a user from tampering with software that controls those features, be it driver workload assessment or an ignition lock, could have grievous results and possibly significant legal ramifications. As an example, software that permitted the user to disable a court-mandated ignition lock, which unlocks the ignition only if the driver has a detected blood alcohol content below the legal limit or none at all, could be argued to be contrary to public good, if not in violation of the initial order requiring the ignition lock. There are at least some circumstances where it is arguably quite reasonable for car companies to not want some of the software in the car to be modified.

# Addressing Anti-Tivoization in Automotive Software

GPLv3 includes a provision that allows a copyright holder to use that license but to include "Additional Permissions" granting additional rights to the licensee::

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law....

You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.<sup>14</sup>

This provision of GPLv3 also allows downstream licensees to remove these additional permissions, if they so desire;

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.)<sup>15</sup>

This provision of GPLv3 provides a mechanism by which a copyright holder who prefers GPLv3 for their code, but is concerned about the effect of the Installation Information requirement on its downstream customers or end users, to grant an additional permission that does not obligate a licensee to follow the Installation Information requirement. At least one project has adopted such an additional permission, which could serve as a template:

The copyright holders grant you an additional permission under Section 7 of the GNU General Public License, version 3, exempting you from the requirement in Section 6 of the GNU General Public License, version 3, to accompany Corresponding Source with Installation Information for the Program or any work based on the Program. You are still required to

<sup>&</sup>lt;sup>14</sup> GNU General Public License version 3.0, Section 7, <u>http://www.gnu.org/licenses/gpl-3.0.en.html</u>

<sup>&</sup>lt;sup>15</sup> *Ibid.* 

comply with all other Section 6 requirements to provide Corresponding Source.<sup>16</sup>

An additional permission under Section 7 of GPLv3 which exempts the licensee from the Installation Information requirement of that license, might allow for GPLv3 software to be used in automobiles while still locking down the software on the head unit to prevent the end user from changing and reinstalling the software.<sup>17</sup>

## Conclusion

GPLv3 compliance in automotive applications may hinge on mitigating the effects of GPLv3 Section 6 and the requirement for sharing of installation information. For many automobile makers, and perhaps the regulatory authorities which set standards for automobiles, the Anti-Tivoization clause of GPLv3 may be considered a deal breaker for reasons of safety. Use of an Additional Permission that exempts the licensee with complying with the Installation Information requirement may be a way to allow for use of GPLv3 in automotive applications while addressing these safety concerns. Other methods, of course, may also exist; the Free Software Foundation (FSF) believes legislation can help.<sup>18</sup> Free Software has the potential not just to play an important role in yet another industry, it has the potential to save lives, quite literally. Once licensing and compliance is understood I think a very strong case can be made that the transparency enabled by FOSS makes safety-critical devices easier to produce, of higher quality, and more effective. This is why there may be the need, at least at this time, to provide a mechanism by which GPLv3 can be used in the automotive industry while addressing their current concerns that the Anti-Tivoization clause may cause safety concerns.<sup>19</sup>

<sup>&</sup>lt;sup>16</sup> E.g., the Canola Project. See Edward T. Lima, "Additional Permissions to the GPLv3", <u>https://garage.maemo.org/forum/forum.php?forum\_id=3771</u>

<sup>&</sup>lt;sup>17</sup> Although Additional Permissions are explicitly allowed in the text of GPLv3, and have been used by projects to exempt licensees from the Installation Information obligation, see, ibid., the use of such an Additional Permission carries risks. First, making use of such a mechanism could require that all code (or at least all code for which it is not desired to provide Installation Information) in the software stack include this Additional Permission - a potentially difficult or impossible task if the stack is complex or requires code from a variety of different projects. There might also be the difficult issue of license incompatibility with code licensed under GPLv3 without such an Additional Permission. If the developer base for the components in the software stack are believers either in the value of the Installation Information requirement, or dislike any effort to alter the "purity" of GPLv3 with Additional Permissions, it may not be possible to make use of this proposal. In addition, any Additional Permission that exists in GPLv3 code may, per Section 7 of GPLv3, be removed by downstream licensees. This could also complicate the creation of a software stack not requiring compliance with the Installation Information requirement. Thus, although this article suggests that an Additional Permission exempting the licensees from complying with the Installation Information requirement might help address some concerns within the automobile industry with GPLv3, the logistics of using and maintaining the Additional Permission might present more complications than the value of the Additional Permission in the first place.

<sup>&</sup>lt;sup>18</sup> See "Volkswagen's Diesel Fraud Makes Critic of Secret Code a Prophet." <u>http://www.nytimes.com/2015/09/23/nyregion/volkswagens-diesel-fraud-makes-critic-of-secret-code-a-prophet.html?\_r=0</u>

<sup>&</sup>lt;sup>19</sup> Many thanks to the members of the Free Software Foundation Europe's safety-critical special interest mailing list and countless others who've helped me with this article.

# About the author

*Jeremiah C. Foster* is an American living in Sweden, works for Pelagicore AB, and prefers Free Software to Open Source.

