

# **BUSMASTER – An Open Source Tool**

**Dr. Tobias Lorenz, ETAS GmbH  
Presented by Dr. Andrew Borg**

In August, ETAS and Robert Bosch Engineering and Business Solutions (RBEI) jointly published BUSMASTER, a free open source PC software for the design, monitoring, analysis, and simulation of CAN bus systems. The software can be downloaded from <http://rbei-etas.github.com/busmaster/>.

The current BUSMASTER version is based on the preceding software tool CANvas, conceptualized, designed and developed by RBEI. It offers import filters for network description files and simulation programs compliant with standard industry formats. For CAN connections, hardware from different vendors is supported.

The BUSMASTER software project, sponsored by RBEI and ETAS, is open to contributions from research and industry. The software can be developed and managed with free software tools. Thanks to the modular architecture, third-party software developers can easily add new functions to the software. The license also permits the provision of proprietary add-ons, which can be dynamically linked to the open source core.

The openness of the project managed by the sponsors provides for flexible modification and extensions regarding bus systems, protocols, and hardware interfaces. In addition, it will facilitate short cycles in the solution's onward development.

ETAS and RBEI (Robert Bosch Engineering and Business Solutions Limited, India) recently published a PC-based rest bus analysis and simulation tool (see Figure 1) on the open source platform GitHub. The major motivation for this is to launch and prototype a new business model in a market with high entry barriers. The project was carried out in two phases: the preparation phase and the operation phase. The preparation phase included the investigation of many Open Source Software (OSS) aspects, e.g., source quality, business models, open source infrastructures, legal implications, customer communication, community buildup, marketing, publishing processes, and organizational aspects. The operation phase starts with the going live on GitHub and includes for example community management, engineering services, and support.

This document describes the challenges of community buildup and technical development.

The project is jointly carried out by ETAS and RBEI. The OSS was conceived, developed, and prepared by RBEI.

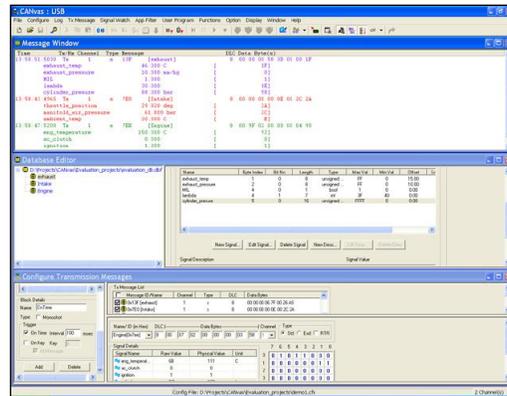


Figure 1: Tool screenshot

## Open Source license

For the ETAS and RBEI bus analyzer tool a LGPL license was chosen. The reason lies in the business model, which shall permit an ecosystem of free or commercial add-ons. These add-ons are dynamically linked to the tool's modular plug-in architecture.

A GPL would require these add-ons to use the same license and would therefore not be consistent with an ecosystem. On the other hand a permissive license would not require anyone to contribute modifications back to the originator and would also not be beneficial for market penetration of the tool.

## Building an OSS community

The attractiveness of a project depends heavily on the first impression of the software. Therefore the barriers for using and developing the software were lowered.

The tool already provides most functions expected from it. However the intuitive and obvious way to use these functions and a well documentation, e.g., a good user manual, tutorials and examples, is always a topic for improvements.

Open Source developers appreciate high quality, well documented source code, and support for cost free build environments and tool chains.

Together user and developer expectations are the major success factors for project attractiveness from a technical perspective [1][2].

From a marketing point of view it is beneficial to use an OSS hosting facility that is already popular with a lot of projects and developers. This is true for SourceForge, which is more project-oriented, and also for GitHub, which is more oriented towards individual users. In the latter case, the user clones complete project repositories from other users, makes modifications and sends pull requests back to the original user. This results in several almost identical repositories among all GitHub users and is one of the reasons

for the huge growth of this platform in the last couple of years. BUSMASTER has chosen GitHub as being the hosting platform.

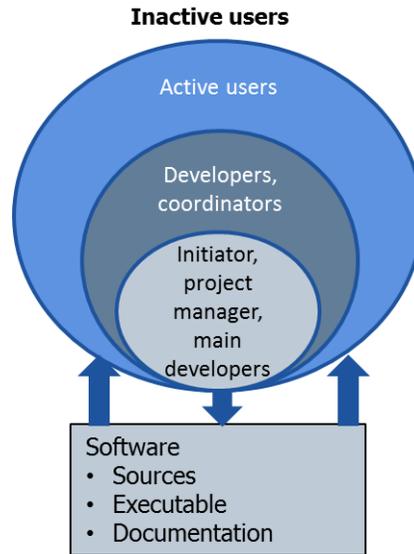


Figure 2: OSS Community

When building an OSS community, the motivation for individuals to it must be evaluated and the project adjusted accordingly. Collaborators are often found in the academic environment. They start using the OSS solution because it is cost attractive. Often they become developers as soon as they make useful adaptations for their own purposes. For them, the motivation lies in communicating with other users and developers and getting the feeling that their contributions are welcome and quickly incorporated in the project's source code. For partner companies the major motivation is business potential.

In case of the ETAS and RBEI rest bus simulation tool, hardware manufacturers are interested in contributing just the required parts (drivers) to make the software work with their own products. The more the software supports their interfaces, the more attractive their complete solutions are on the market. This is the specific motivation for them to contribute to the project.

If the project and the number of developers contributing to it grow, a project structure becomes necessary. This usually consists of a steering committee for the long term strategic goals of the project. If necessary, smaller groups can be setup to focus on specific technical aspects, e.g. driver support, interoperability, usability, and accessibility. Usually the more a developer is recognized by his project contributions, the more influence he gets on the long term development. This process of self-organization is referred to as having "meritocratic government and rights" among all participants and nowadays is often the focus of social research.

## **Issue tracker**

GitHub provides an Issue Tracker. The BUSMASTER project extensively makes use of it to collect new ideas for future development. Anyone is able to raise issues and take part in the discussions. The Issue Tracker not only contains feature requests, but also broken down tasks and bug reports.

Every issue is marked with one label per category:

- 1 (version): bug fix, major or minor
- 2 (system): bus system, file format, hardware interface, other, transport protocol
- 3 (progress): in progress

Every issue can be assigned to a developer. Thus he claims to work on them.

Every issue can be assigned to a certain milestone. Thus he can commit to a delivery date.

The Issue Tracker already contains several areas interesting for research and development and is open for further topics.

## **Panel support**

An often asked feature is a user-defined panel to overlay the BUSMASTER window. Thus own display elements and input methods could interact with events like signal transmit or receive.

A panel designer could be provided that stores the panels in an XML-format.

An import filter could be provided at least for the Vector panel files.

## **Multi-platform support**

Multi-platform support is another important issue. For this BUSMASTER must be migrated from today's C++/MFC to a multi-platform toolkit, e.g. Qt or GTK+.

The goal is to run BUSMASTER under Linux and ideally Android or iOS.

For Linux, distribution specific packages could be provided (Debian/Fedora/OpenSUSE or Ubuntu/RedHat/SUSE). At the end BUSMASTER could find its way in all big Linux distributions.

## **User interface improvements and wizard**

Restructuring and redesigning the user interface and adding wizards for the most common use cases could improve the intuitive usage. This is not only area for computer scientists, as analysis and optimization is based on test user experience. As this difference among the users, an option to select the user interface complexity during installation and runtime is good way to cope with it.

## **Support for bus system**

BUSMASTER currently only support CAN bus communication. However there is a big interest to also support Ethernet, FlexRay, LIN and probably digital sensor buses like PSI5, for which tools are nonexistent yet.

Introducing new bus systems includes specific hardware abstraction, controller configuration and making existing analysis windows more generic.

## **Support for transport protocol**

In each application domain multiple specific transport protocols are used based on CAN. To become successful in these areas, BUSMASTER could provide support for them. Given an example for the automotive domain, this could be XCP/CCP and in the industrial domain CANopen.

## **Support for hardware interface**

At least 30 different companies are offering hardware interfaces that could potentially be supported with BUSMASTER. Implementing the support either for an existing hardware family or adopting BUSMASTER for an own hardware interface could also be an interesting topic.

## **New use cases and software interfaces**

Most innovations take place when different technologies are combined with each other. Provide new methods to work with BUSMASTER, e.g. new analysis or visualization methods, could therefore be in the focus of research.

However the development is always necessary in order to provide interfaces to existing hardware and software products, e.g. data mining (RapidMiner) or ETAS INCA.

## **Summary**

ETAS and RBEI are among the first business units within the Bosch group to publish and lead a new Open Source project. As well as the main interest in market penetration, Open Source is elaborated as a new business model. The risks of this strategy have been evaluated, however, it remains to be seen whether the publication of a formerly commercial product under an Open Source model will lead to increased attractiveness and business opportunities. In any case, as the first such program in this application field it will be in the focus of the Open Source community and will appear in different media. From a business perspective opening an Open Source product equally touches on technical, legal, and

marketing topics. The decisions and experiences made during this project will be guidance for further upcoming Open Source involvements of the Bosch group.

## **References**

1. Open Source community building: a guide to getting it right, URL:  
<http://www.visionmobile.com/blog/2011/01/open-source-community-building-a-guide-to-getting-it-right/>, last viewed: 2011-08-10
2. Open Source Community Building, URL:  
[http://www.opensourcejahrbuch.de/download/jb2006/chapter\\_05/osjb2006-05-03-stuermermyrach.pdf](http://www.opensourcejahrbuch.de/download/jb2006/chapter_05/osjb2006-05-03-stuermermyrach.pdf), last viewed: 2011-08-10