

# Demo: OpenC2X – An Open Source Experimental and Prototyping Platform Supporting ETSI ITS-G5

Sven Laux, Gurjashan Singh Pannu, Stefan Schneider, Jan Tiemann,  
Florian Klingler, Christoph Sommer and Falko Dressler

Distributed Embedded Systems, Heinz Nixdorf Institute and Dept. of Computer Science, Paderborn University, Germany  
{gurjashan.pannu, stefan.schneider}@ccs-labs.org, {slaux, jantie}@mail.upb.de,  
{klingler, sommer, dressler}@ccs-labs.org

**Abstract**—Vehicular networking is at the corner from early research to final deployment. This phase requires more field testing and real-world experimentation. Most Field Operational Tests (FOTs) are based on proprietary commercial hardware that only allows for marginal modifications of the protocol stack. Furthermore, the roll-out of updated implementations for new or changing protocol standards often takes a prohibitively long time. We developed one of the first complete Open Source experimental and prototyping platform for vehicular networking solutions. Our system supports most of the ETSI ITS-G5 features and runs on standard Linux. New protocol features and updates could now easily be done by and shared with the vehicular networking R&D community.

## I. INTRODUCTION

In recent years, different standards for wireless communication in Intelligent Transportation Systems (ITS) have been developed in North America (IEEE WAVE), Europe (ETSI ITS-G5), and Japan (ARIB T109) [1]. Amongst these standards, the ETSI ITS-G5 is currently the only one which incorporates Decentralized Congestion Control (DCC) to adapt protocol behavior based upon changing vehicle densities to avoid overloading of the wireless channel [2]. Nevertheless, the standardization process is not yet complete; new protocol variants are under development [3], [4]. Examples include the introduction of congestion control mechanisms in the WAVE protocol stack as well as the prioritization of messages in combination with congestion control. Typically, these new developments have first been investigated using simulation. However, only Field Operational Tests (FOTs) and experimentation help to validate simulation models and, of course, to reveal limitations of the proposed protocols.

Commercial, and thus proprietary, experimentation equipment includes, for example, the Wireless Safety Unit (WSU) by Denso and the MK5 by Cohda Wireless. However, these systems are rather inflexible when it comes to updates to the protocol stack or the roll-out of updated implementations. As of now, there are some first works towards Open Source implementations, all being focused on certain protocol aspects and not covering a full ITS protocol stack (cf. Table I).

We fill the gap by providing the first Open Source experimental and prototyping platform, OpenC2X,<sup>1</sup> supporting most ETSI ITS-G5 features and running on standard Linux systems.

<sup>1</sup><http://www.ccs-labs.org/software/>

Table I  
COMPARISON OF OPEN SOURCE IMPLEMENTATIONS

ETSI Standard	OpenC2X	GeoNet Stack <sup>2</sup>	Vanetza <sup>3</sup>
DCC 102 687	✓	–	✓
CAM 102 637-2	✓	✓	–
DENM 102 637-3	✓	✓	–
LDM 102 863	✓	–	–
GeoNet/BTP 102 636-4-2/-5-1	–	✓	✓
Security Management	–	–	✓
GPS	✓	✓	✓
OBD-II	✓	–	–

## II. OPENC2X

Figure 1 shows the high-level architecture of OpenC2X realizing the ETSI ITS-G5 stack. As can be seen, OpenC2X consists of highly independent modules, which can be easily extended and reused. In our implementation, all modules are communicating with each other via ZeroMQ. OpenC2X focusses on network functionality but it can be extended further to include GeoNetworking and security. Each module can be started or stopped without affecting the overall system stability.

### A. Access Technology

Wireless transmission is handled by a modified Linux kernel accessing an Ath9k based wireless card. The system runs on Linux kernel 3.18, which is enabled with Outside the Context of a BSS (OCB) mode to support IEEE 802.11p [5]. The core of the networking functionality is the DCC module, handling all incoming and outgoing messages. DCC handles the access to the wireless channel and is responsible for congestion control. It operates in three states: relaxed, active, and restricted and switches depending on the current channel busy ratio. The Ath9k chipset continuously measures this ratio, which is known to DCC by periodic probing via the Netlink interface. For the outgoing packets, DCC checks the type of the packet, i.e., Cooperative Awareness Message (CAM) or Decentralized Environmental Notification Message (DENM). Based upon the type, the packets are added to different access categories supported by IEEE 802.11p and EDCA. DCC requires support for transmit rate control, transmit power

<sup>2</sup><https://github.com/alexvoronov/geonetworking>

<sup>3</sup><https://github.com/riehl/vanetza>

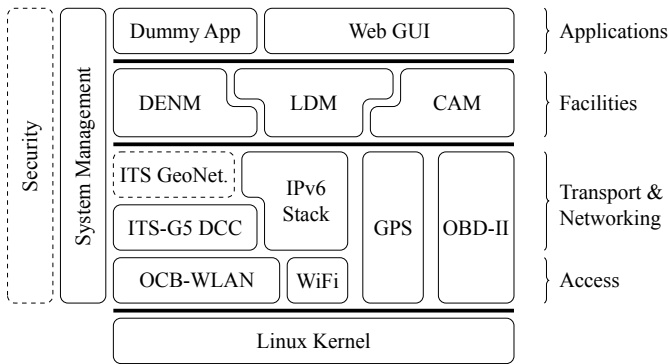


Figure 1. High-level architecture of OpenC2X realizing ETSI ITS-G5 stack. The modules communicate with each other via ZeroMQ. Dashed modules are not yet implemented in OpenC2X.

control, transmit data rate control, flushing outdated packets from hardware queues; their enforcement certainly depends upon the underlying hardware capabilities. OpenC2X currently supports CAM and DENM messages.

### B. Facilities

In the facilities layer, the CAM and DENM modules deal with the respective messages in the ETSI ITS-G5 stack. The Local Dynamic Map (LDM) module manages data from local sensors as well as such received via CAM and DENM messages from neighboring vehicles. CAMs are triggered periodically based on the set of rules specified in the standard, sharing the vehicle's position and state for cooperative awareness. Triggering occurs when position, speed, or heading change significantly, but at least once per second. DENMs can be triggered by connected applications, e.g., for collision avoidance. In our current prototype, triggering can be done manually via the web interface or a connected DENM application. The LDM handles all generated and received data. It stores incoming and outgoing CAMs and DENMs as well as network information (channel load, DCC state, etc.) in an SQLite database. This data is later used by the web interface for real-time monitoring and can also be accessed post-experiment for offline analysis. To improve the performance, latest data is stored in a cache.

### C. Sensors

Both CAMs and DENMs are supposed to include local sensor information. OpenC2X currently supports GPS and OBD-II sensors providing a variety of information. The Linux `gpsd` is used to connect to a GPS device and collects periodic position updates. In order to perform reproducible experiments, previously recorded GPS data can also be fed into the application in form of a trace. Vehicle speed is retrieved by connecting to the vehicle's on-board network using OBD-II.

### D. System validation

We tested the system using (1) PC Engines alix 3d3 system boards, each equipped with a Compex WLM200N5-23ESD wireless miniPCI card using an Atheros AR9220 chipset; (2) PC Engines apu 1d4 system boards, each equipped with two Compex WLE200NX miniPCIe cards using an



Figure 2. Hardware setup for the field test: OpenC2X is running on a small Linux PC in the car; for the GPS receiver and the WLAN card one antenna each is mounted on the roof.

Atheros AR9280 chipset; and (3) a dSPACE MicroAutobox embedded PC, which is a commercial shock and vibration proof prototyping system equipped with a SparkLAN WPEA-121N wireless miniPCIe card using an Atheros AR9382 chipset. Given the variety of the PC hardware and the different Atheros chipsets, the performance of the underlying hardware is thus comparable to other commercial tools. Even better, the system provides a wider degree of flexibility.

To validate OpenC2X, we confirmed inter-operability with a commercial Cohda MK5 and were able to send and receive both CAM and DENM messages. We further performed a small field test with two cars and two Roadside Units (RSUs). Figure 2 shows one of the cars equipped with OpenC2X: A u-blox GPS receiver and an ELM 327 OBD-II were used to receive the sensor data. During the experiment, we successfully received the periodic CAMs and the manually triggered DENMs of all other stations within communication range.

## III. CONCLUSION

We presented our novel OpenC2X system, an Open Source experimental and prototyping platform supporting the ETSI ITS-G5 standard. It complements already existing implementations that lack important functionality such as DCC or LDM handling. OpenC2X enables real-world tests and validations of new protocol options and standards. By making OpenC2X Open Source, we hope for future contributions to further improve its performance and compliance to the standards. Given its highly extensible architecture, novel modules including GeoNetworking along with security aspects can be contributed by the research community.

## REFERENCES

- [1] C. Sommer and F. Dressler, *Vehicular Networking*. Cambridge University Press, Nov. 2014, p. 384.
- [2] ETSI, "Intelligent Transport Systems (ITS); Decentralized Congestion Control Mechanisms for Intelligent Transport Systems operating in the 5 GHz range; Access layer part," ETSI, TS 102 687 V1.1.1, Jul. 2011.
- [3] G. Bansal, B. Cheng, A. Rostami, K. Sjöberg, J. B. Kenney, and M. Gruteser, "Comparing LIMERIC and DCC Approaches for VANET Channel Congestion Control," in *IEEE WiVec 2014*, Vancouver, Canada: IEEE, Sep. 2014, pp. 1–7.
- [4] C. Sommer, S. Joerer, M. Segata, O. K. Tonguz, R. Lo Cigno, and F. Dressler, "How Shadowing Hurts Vehicular Communications and How Dynamic Beaconing Can Help," *IEEE Transactions on Mobile Computing*, vol. 14, no. 7, pp. 1411–1421, Jul. 2015.
- [5] R. Lisovy, M. Sojka, and Z. Hanzálek, "IEEE 802.11p Linux Kernel Implementation," Czech Technical University, Prague, Czech Republik, Technical Report, Dec. 2014.