Journal of Mobile Multimedia, Vol. 10, No.1&2 (2014) 128-140

AN IN-VEHICLE EMBEDDED SYSTEM FOR CAN-BUS EVENTS MONITORING

SHICHAO CAI

China HUANQIU Contracting & Engineering CORP.
caishichao@gmail.com

MOHAMED BAKHOUYA

International University of Rabat, Morocco
mohamed.bakhouya@uir.ac.ma

MOHAMED BECHERIF

FCLab FR CNRS 3539, Femto-ST UMR CNRS 6174

mohamed.becherif@utbm.fr

JAAFAR GABER and MAXIME WACK

SET, University of Technology Belfort-Montbeliard, 90010 Belfort Cedex, France

{gaber, maxime.wack}@utbm.fr

ITS applications usually involve inter-vehicle communication as a medium to exchange information between vehicles experiencing undesirable and/or dangerous situations (e.g., traffic jam, accident, or bad road conditions) and other surrounding vehicles. For example, if the in-vehicle ABS is activated, it might indicate bad road conditions, and if this information is shared between vehicles, other drivers will be warned to take preventive actions before getting into a dangerous situation. In this paper, an embedded system is presented for in-vehicle event monitoring. The system uses a diagnostic interface to plug into the vehicle's existing network (i.e. CAN bus) and a software is integrated to extract relevant information. Wireless communication modules, mainly WLAN, ZigBee and GPS/GPRS, are integrated into the system to communicate with other surrounding vehicles as well as emergency/control centers. A prototype is developed and deployed in a real vehicle, experiments are conducted, and results are reported to show the efficiency of the system and the soundness of gathered in-vehicle data.

Key words: In-Vehicle embedded systems, Inter-Vehicle communication, ITS context-aware applications, On-board sensors and CAN bus.

1 Introduction

In harsh environments, it is hard for drivers to anticipate dangerous situations on the road that could cause serious accidents, especially when driving during the night or in the fog, due to the blurring of vision, or a high speed driving. Moreover, inappropriate tire pressure can also compromise vehicle stability, its handling and braking, and could contribute to an accident. With the growth and expansion of wireless communication and embedded technologies, considerable research efforts have been done in the area of Inter-Vehicle Communications (IVC) [4]. The objective is to increase vehicle safety and driver comfort by relaying required information from vehicle to vehicle. Two types of applications can be distinguished, comfort applications and safety applications [4]. Comfort applications can improve passenger comfort by updating on route optimization and increasing traffic efficiency e.g., weather information, gas station or restaurant location. Safety applications increase the passengers' safety by exchanging information via inter-vehicle communication. For example, a vehicle detecting an icy road could inform other vehicles intending to use the affected route.

In order to gather information about the driving environment, the potentials for in-vehicle systems are currently investigated. Most of existing systems are based on centralized architectures and fixed infrastructures on roads (cameras, sensors), which incur a high deployment cost and the reaction/response time could be long due to information transmission and processing. In addition, infrastructures on roads need continuous maintenance. Therefore, systems with the minimum deployment cost and the ability for efficient information transmission and processing are required [6]. Hence, to tackle this issue, collaborative driving techniques have been recently investigated using inter-vehicle communication technologies. Several projects[a], such as COOPERS (CO-operative Networks for Intelligent Road Safety), SAFESPOT (Cooperation between intelligent roads and vehicles), CVIS (Co-operative Vehicular Infrastructure Systems), and PreVENT, were launched to develop collaborative driving techniques.

The long term vision of these projects is to develop techniques and approaches allowing vehicles to anticipate and avoiding collisions, navigate the quickest route to their destination, making use of up-to-the minute traffic reports, identify the nearest available parking slot and minimize their carbon emissions. For example, the COOPERS project emphasizes on the mutual information exchange between vehicles and road side units in order to provide better safety related information to drivers. Roadside architecture is defined for data acquisition, applications for the traffic control center, roadside transmitters for V2I communication, an onboard unit that supports different kinds of communication technologies, and information services. SAFESPOT, with the assistance of the Car-to-Car Communication Consortium (C2C-CC), proposes an innovative solution that enables V2V and V2I communication. The underlying concept is known as Vehicular Ad-hoc Network (VANET) that uses Digital Short Range Communication (DSRC) defined by IEEE 802.11p to fulfil communication needs [3]. In SAFESPOT, several safety functions have been proposed to anticipate potentially dangerous situations and to inform drivers about these risky conditions. The CVIS project aimed to offer a unified technical solution for communication between vehicles and infrastructure elements using a variety of access technologies and to enable a wide range of potential cooperative applications for vehicle and roadside equipment by defining an open and scalable platform.

The work presented in this paper is part of the work done in the EU FP7 project, ASSET (Advanced Safety and Driver Support for Essential Road Transport)[b] , where the main objective was to develop applications to enhance driver safety. Here the focus is only on extracting, recording, and sharing information (i.e., in-vehicle data from the CAN bus), between neighbouring vehicles. To

[a] http://www.coopers-ip.eu/; http://www.safespot-eu.org/; http://www.cvisproject.org/; http://www.prevent-ip.org/
[b] http://www.project-asset.com/

achieve this objective, an embedded system was developed to allow extending drivers knowledge about the surrounding environment and warning them of undesirable road conditions. More precisely, the embedded system allows vehicles, within a geographical area, to communicate and share their driving experiences in order to either confirm or reject emergency situations [9]. For example, whenever a vehicle perceives an abnormal road condition (e.g. slippery road condition), this information should not only be conveyed to the local car driver but could also be propagated to other vehicles in the same neighbourhood, so they pay attention and avoid potentially dangerous situations. However, information coming from only one vehicle may not be credible and reliable on which to trigger a confident action. Involving multiple vehicles in exchanging information will increase the confidence of a global current driving situation. For example, once the system is notified that the ABS (Anti-Lock Braking System) of the vehicle is activated to indicate a potentially dangerous situation such as an incident or bad road conditions (e.g., icy road, strong rainfall or snow), this information are shared and other drivers will be informed to take preventive actions before getting into the same dangerous situation [3]. Furthermore, recording information and storing them in a database will help viewing and analyzing driver behavior and vehicle trouble that might have caused accidents.

The embedded system, presented in this paper, was designed and implemented for information monitoring and uses a diagnostic interface to plug into the internal network of the vehicle (CAN bus). The main objectives are twofold. First, gathered events will be analyzed and used for warning the driver before getting into dangerous situations. Second, this information will be also used to provide, at any desired time, a substantial record of the different events that have occurred while driving. Recorded data can be, for instance, used for analyzing the causes of accidents and driver behaviors. Data are temporarily stored in an industrial and compact embedded computer memory and older ones are discarded to avoid memory overflow. All recorded events are uploaded into the system in the infrastructure for filtering, processing and storage for eventual usage (e.g., accidentology analyzing purpose or driver behavior analysis [13]). To achieve these objectives, a data-logger software was developed to extract events occurred. Wireless communication modules, mainly ZigBee and GPS/GPRS, are integrated within the system to communicate with other surrounding vehicles as well as the emergency/control center. Hence, the embedded system is able to connect into the existing CAN bus system of light-duty vehicles and decode exchanged messages in order to collect some information about the vehicle internal state.

The remainder of this paper is as follows. Section 2 presents an overview of CAN buses. In Section 3, the architecture of the in-vehicle embedded system is presented. Section 4 presents the data-logger software for internal event decoding. In Section 5, the experimental studies together with experimental results are presented. Conclusions and future work are presented in Section 6.

2 CAN Bus Overview

The modern vehicle is no longer a single assemblage of components operating independently of each other. There are more and more electronic components being used in-vehicle in order to make it more safe and comfortable and to satisfy the government's requirements for improved emission control and reduced fuel consumption. Examples of such components include engine management, active suspension, ABS, gear control, lighting control, air conditioning, airbags and central locking. The increasing number of components and their complexity requires a communication interface and the

CAN bus has become the most efficient way to enable vehicle's electronic components to communicate with each other [2].

CAN is a serial communication network that uses Non Return to Zero (NRZ) encoding (with bit-stuffing) for data communication on a differential two wire bus CAN-High and CAN-Low (CAN H and CAN L) [1,8]. Almost every new car manufactured is equipped with at least one CAN. An example of the distributed control architecture of the Volvo XC90 is given by [11]. Bosch developed the CAN, which has been internationally standardized (ISO 11898) since 1993 and has been "cast in silicon" by several semiconductor manufacturers. It is widely used in the automotive field to make vehicles electronic components interact with each other. CAN has been the subject of a lot of research and development. For example, in [10] a CAN was designed to meet the need of real-time control for the traction control system (TCS) of a 2WD vehicle. A network control system of CAN for electronic control and detection system for a dry hybrid belt with continuously variable transmission (A-CVT) is developed [8]. In [1,12,14], authors have studied features of CAN buses and designed many applications.

However, to the best knowledge of the authors, few works have been done for monitoring information inside the CAN bus system. A vehicle's existing CAN bus provides numerous important information, such as information on Airbag, on the trigger of ABS, about the engine, etc. This information can be exploited and used to develop applications in order to increase road safety. However, in light-duty vehicles, existing CAN buses are commonly private and closed. Every automobile manufacturer has its own CAN bus message protocol, identifier and codes. Therefore, identifiers of exchanged message frames cannot be directly decoded because they are hidden by vehicles manufacturers. Hence, information from the vehicle CAN bus cannot be directly accessible through this system.

The embedded system for event monitoring introduced in this paper is able to interact with the CAN bus system, extract relevant information and send it to the infrastructure or to other vehicles. In other words, the objective is to extract relevant information and warn the driver about dangerous situation by offering a support for inexperienced drivers in a harsh or unfamiliar environment.

3 Embedded system architecture

In this section, an embedded system for gathering relevant events like the trigger of ABS is presented. The in-vehicle system could warn the driver in a dangerous situation and offer support for inexperienced drivers. For example, driving at night or in an area with fog, due to the blurring of vision and a high speed of driving, it is difficult for drivers to detect an obstacle on the road. Incorrect tire pressure can compromise the stability of a vehicle, its handling and braking, in extreme cases, could contribute to an accident. Therefore, if drivers are informed or warned by such information like the pressure of their vehicle tires and road related events (e.g., object on the road, fog), many accidents can be avoided. Currently, there are several relevant information, remaining inside vehicles, that could be extracted from the CAN bus. If this information is shared between drivers of vehicles situated in the same neighborhood, new knowledge will be acquired and drivers could improve their decision making capabilities by taking actions before getting into dangerous situations. Information can be passed to the drivers as a warning or used for active intervention on the driving process.

The architecture of the in-vehicle embedded system is composed of the following components: a diagnostic interface, an embedded computer, a wireless communication and GPS component, and a touch screen (see Figure 1). The used embedded computer is a CompactRIO (cRIO) programmable automation controller (PAC) with a low-cost reconfigurable control and acquisition embedded system designed for applications that require high performance and reliability [7]. The system combines an open embedded architecture with small size, extreme ruggedness, and hot-swappable industrial I/O modules. NI CompactRIO consists of a real-time controller, a reconfigurable FPGA, a reconfigurable chassis, a deterministic Ethernet expansion chassis, and I/O modules. The programming of the cRIO is done by the Labview development software.

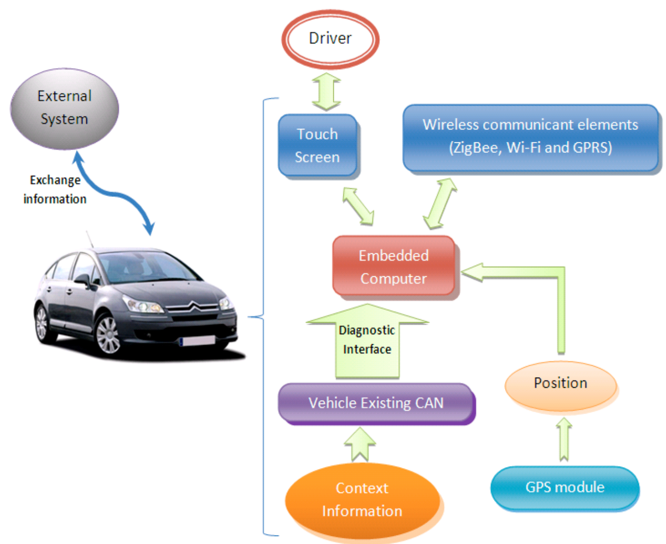

Figure 1 Block diagram of the designed system

A diagnostic interface is needed to plug into the vehicle's existing CAN bus. cRIO is used for acquiring CAN frames coming from the vehicle diagnosis plug, and programmed with the Labview development software for extracting useful frames, transferring the enforceable files compiled under Labview into the FPGA and the real-time controller of the cRIO. ZigBee is used for the wireless communication between vehicles (V2V), and the GPS/GPRS module for wireless communication with an infrastructure (V2I) since it offers the vehicle's position and supports a long-distance transmission [4]. A GPS module made by SEA-gmbh for the cRIO is used to acquire in real-time the vehicle location. This module is called SEA cRIO Gxxx+. The high sense GPS receiver in the Gxxx module allows the reception of the US GPS system for positioning and timing purposes. The modules deliver worldwide position information, accurate time and time zone information. A touch screen is integrated into the system for interacting with the driver. The NI TPC2106 Touch Panel is used and is working under Windows CE and programmed with LabVIEW.

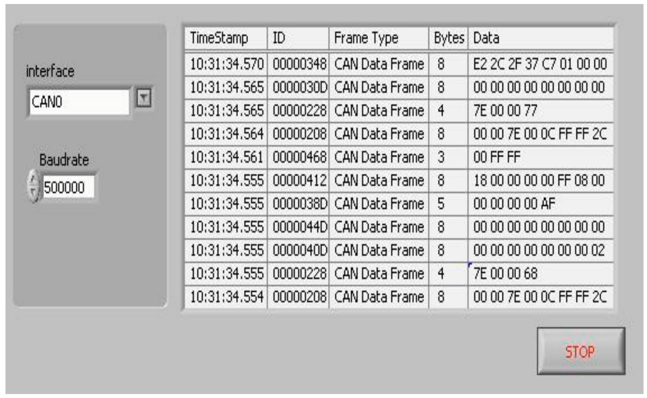

Figure 2 The received frame

For the CAN frame acquisition, the National Instruments NI 9853 CAN module is selected. It can work with either standard 11-bit and extended 29-bit CAN arbitration ID’s. It is also able to transmit received CAN frames with a 100 percent bus load at 1 Mb/s. Thus, the computer can gather information from the vehicle's CAN as described in Figure 2. Note that there are several CAN Buses in the vehicle e.g. the security CAN bus and the comfort CAN Bus. The vehicle's existing CAN bus related to the safety is connected with the OBD II connector.

4 Internal context decoding

The data-logger software is developed using LabVIEW for extracting, decoding, and saving the frames coming from the vehicle CAN bus. It is an FPGA program operating as follows: read CAN frames to arrive, checks out errors and if they exist, transfers the error code and data as outputs. If there is an error with the code 65539, meaning that the input missed one or more data (in which case it is recommended to make sure that the loop can run as fast as the module data rate), the error is displayed through a LED. Afterwards, data are transferred into a FIFO stack if there are no errors. Otherwise, the CAN module is reset. If the FIFO is full, the VI waits up to 100 ms for the host program to dequeue a frame from the FIFO, before it discards the data. The FIFO stack will be read by the host program running on the cRIO.

Table 1 shows the output of the developed LabVIEW program, which displays the received CAN frames. Other information like the time, identifiers, the frame type, the data length and content of data are also displayed. A baud rate of 500kb/s is needed in order to receive frames correctly. This value was obtained through many trials.

The next step is to identify the CAN bus frames (i.e. finding identifiers), filter, and collect interesting frames representing the internal information about the vehicle state. The program reads the data from the FIFO stack and treats them in order to break them down into IDs, BYTES, DATA, and NUMBER of bytes (of the data frame with a maximum value of 8). It is worth noting that identifiers are hidden information by automotive companies and difficult, or even impossible, to get them from car manufacturers. The decoding was successfully done after many trials for finding a change in the data of frame according to each identifier. Every independent operation would change some frame data. It was therefore necessary to repeat actions (e.g., switch on lights crossing) to identify changes in bit-level of the corresponding frame.

ID	Octet	Bit	Descr.	Code
228	3	1-8	State of Accel. Pedal	Min :00 Max :FF
412	1	3	Reverse	No active :0 Active :1
412	1	4	Hand break	No active :0 Active :1
412	1	6	Break	No active :0 Active :1
432	6	4	Driver belt	No active :0 Active :1
612	2	7	Left light	Turn off :0 Turn on :1
612	2	8	Right light	Turn off :0 Turn on :1
612	2	1	Night light	Turn off :0 Turn on :1
612	2	2	Code light	Turn off :0 Turn on :1
612	2	3	Headlight	Turn off :0 Turn on :1
612	2	4	Fog light	Turn off :0 Turn on :1
488	6	1-8	Engine temp.	Min: -40°C Max: 214°C
38D	1-2	1-8	Speed	Min: 0 km/h Max: 656 km/h
208	1-2	1-8	Engine Speed	Min: 0 tr/min Max: 8191.75 tr/min
50D	6	6	ABS	Off: 0 In regulation:1

Table 1 Identifiers table

5 Experiments and results

Experiments were conducted with two vehicles for message exchanges using ZigBee and GPRS communication technologies and considering the ABS activation alert. The test scenarios, together with the system architecture, are illustrated in Figure 3. Vehicle 1 triggers its ABS during the test and information of the ABS activation events was extracted from the CAN bus via our event recording system. It stores the different information to determine if some information is crucial for driving safety and to decide in displaying that information to the driver on the touch screen or sending them through wireless devices either to vehicle 2 or to the infrastructure (e.g. emergency or control center). The control center can monitor vehicle position, speed, ABS activation, etc. By using this system, vehicles within a geographical area should be involved by communicating relevant information in order to either confirm or reject an emergency situation. For example, if the control center receives several messages of ABS triggering, then it will call the corresponding service to handle the problem before a driver gets in a dangerous situation.

5.1 Direct alert of ABS's activation

The experiments of the direct ABS’s activation alert were conducted in an urban area and in a highway of the city of Belfort (France). For data transmission, XBee-PRO 868 modules from Digi were used. These XBee modules have the same PHy and MAC layer as the ZigBee standard (IEEE 802.15.4), and therefore are ZigBee-compliant. They were chosen for testing because they offer long range embedded

RF modules for European applications (868 MHz short range device G3 band for Europe), reliable transmission, 128-AES encryption and API support, and allow the use of peer-to-peer or point-to-multipoint topology. For GPS coordinate recording, GPS receivers, from Navibe GM720, designed with the SiRF Star III (GSC3f Base band processor with integrated flash memory and RF front end) single chipset were used. By using SiRF Star III single chipset, GPS-mouse produces high sensitivity to satellite signals with low power consumption. It can track up to 20 satellites at a time and update data position every 0.1 second. A program was developed to allow the reception of data from the USB GPS receiver, analyze GPS information, periodical data transmission through the XBee-PRO 868 module. For each transmission, the program records the received signal strength (going from -63 dBm to -112 dBm) to check if data have been correctly received.

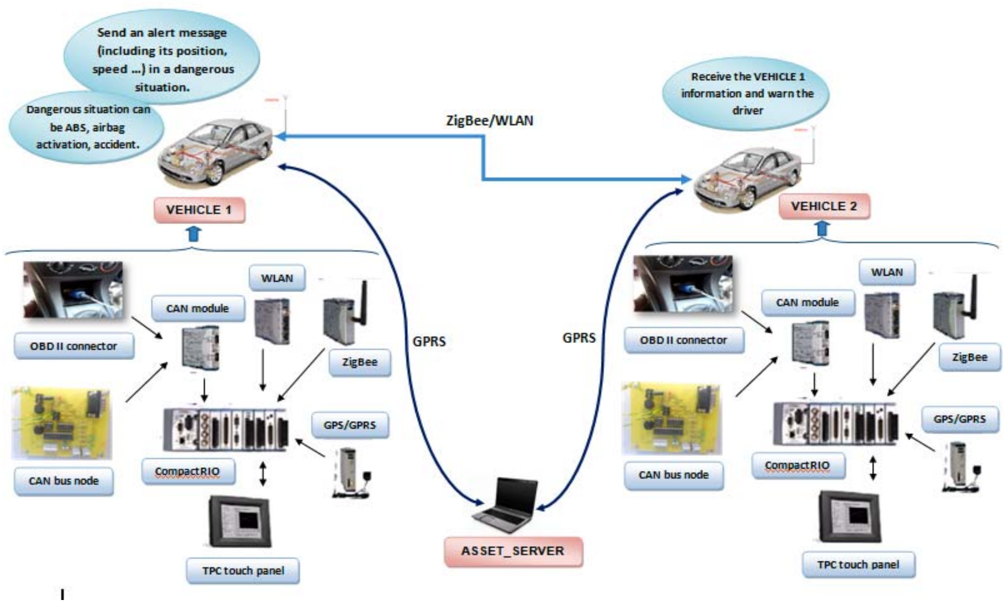

Figure 3 ABS activation alert experimental system and the test scenario

Figure 4 shows the itinerary followed in both the urban setting and in a highway. Several tests were conducted with different vehicle speeds and distances. When the ABS of vehicle 1 was triggered, the driver was warned through the touch screen and at the same time, this information was transmitted successfully to other vehicles in the vicinity (e.g., vehicle 2) by a ZigBee wireless communication module. Vehicle 2 received instantaneously the information about the triggering of the first vehicle ABS activation as shown in Figure 5. It should be noted that the transmission distance of ZigBee is a maximum of 300m in an urban setting. Vehicle 2 received the warning message within the risky area. Actually, this radio distance limitation is not a drawback because the aim is only to warn vehicles in near vicinity. Regarding this distance and the urban area (with speed limitation), the warned drivers has largely time to adjust their speed and eventually take preventive actions (e.g., re-routing to bypass the indicated area).

(a) Urban (b) Highway

Figure 4 Itinerary used during the experiments

(a) Vehicle 1: sender (b) Vehicle 2: receiver

Figure 5 Context information about the vehicle from the vehicle existing CAN system

It's worth noting that, according to conducted experiments, the signal reception quality is not affected by the speed variation of both vehicles but is only affected by the relative vehicle distance. According to the specification of the used ZigBee module, the maximum distance for receiving signal is around 300m. Results presented in Figure 6 illustrate the relation between the signal level and the distance between the two vehicles in the city. When the distance between the two vehicles grows the signal level decreases. Figure 7 shows the relation between the signal level and the distance between the two vehicles on the highway. As there are fewer obstacles on the highway, the communication range is higher (between 750m and 1Km). For security applications, this fact is very useful, highway users could be informed sooner, mainly in case of an accident.

Another objective of these experiments was to study the soundness of the real-time data extracted from CAN bus. Figure 8 shows some information about the vehicle internal context obtained by the deployed system. Figure 9 shows the logical relationship between brake events and vehicle speed. As a brake event occurs, the vehicle speed changes. One can also see that the brake and speed variation are in the same order of magnitude with similar trend. Figure 10 shows the coherency between acceleration events, vehicle speed and engine speed. As the acceleration value increases, one can see a simultaneous increase in vehicle speed and engine speed with exact accordance with the timing of events.

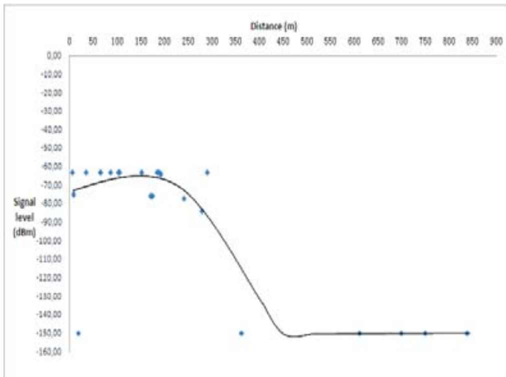

Figure 6 Signal level VS distance in urban

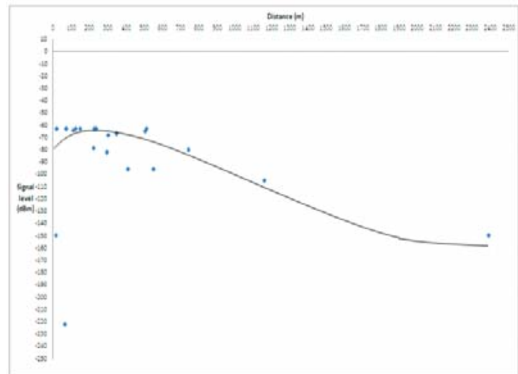

Figure 7 Signal level VS distance on the highway

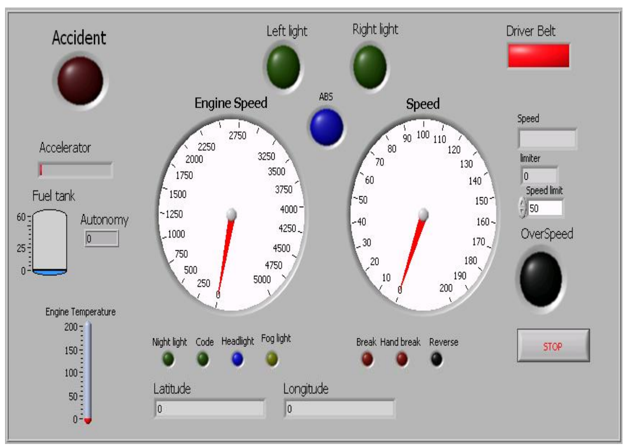

Figure 8 Context information about the vehicle from the vehicle CAN bus

5.2 Indirect alert of ABS's activation

The information of the ABS activation was also delivered to a server of an infrastructure (ASSET-Server) by a GPRS wireless communication [5]. When the ABS is triggered, the CAN transmits the information to the Labview module. The Labview module should send messages to the ASSET-Server. There are two kinds of messages from a vehicle: “GPS” to inform the Server of its coordinates and “ABS” to alert the Server of a risk. After that, ASSET-Server redirects these messages to the ABS Server module, considered a client for the ASSET-Server. Their functions are described as the following: save car positions if the “GPS” message is received. Otherwise, the “ABS” message will trigger events to look for the nearest vehicles to the point where the “ABS” was reported, and going in the same direction as well. The ABS Server sends “WRN” messages to alert the right vehicles of a risk (i.e. figure 11). Finally, drivers are informed through the warning message displayed on the touch screen.

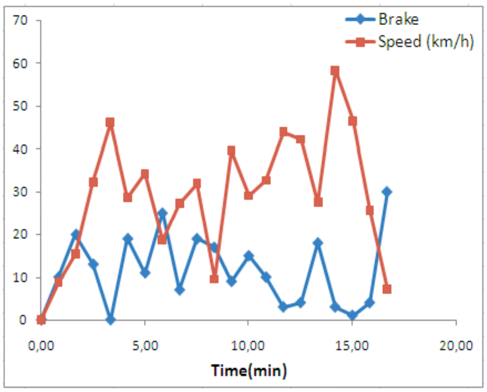

Figure 9 Brake vs. Acceleration

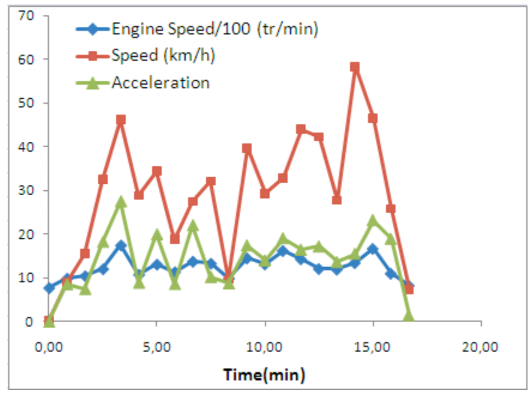

Figure 10 Acceleration vs. engine speed and vehicle speed

An experimental test was performed on a defined route in order to send the alert of ABS activating in some zones or receive WRN messages. In Figure 12, the blue squares correspond to the ABS alerts recorded by the Labview module. The black points correspond to the ABS alerts recorded by the ASSET-Server. One can conclude that each ABS triggered by the experimental vehicle is surrounded by a blue square. This means that each ABS alert sent by the Labview module was correctly received by the ASSET-Server. Of course, other black points can be observed, not surrounded by a blue square corresponding to ABS alerts from other vehicles. WRN messages received by vehicles are presented by green squares. It can be seen that each warning received is close to an ABS alert.

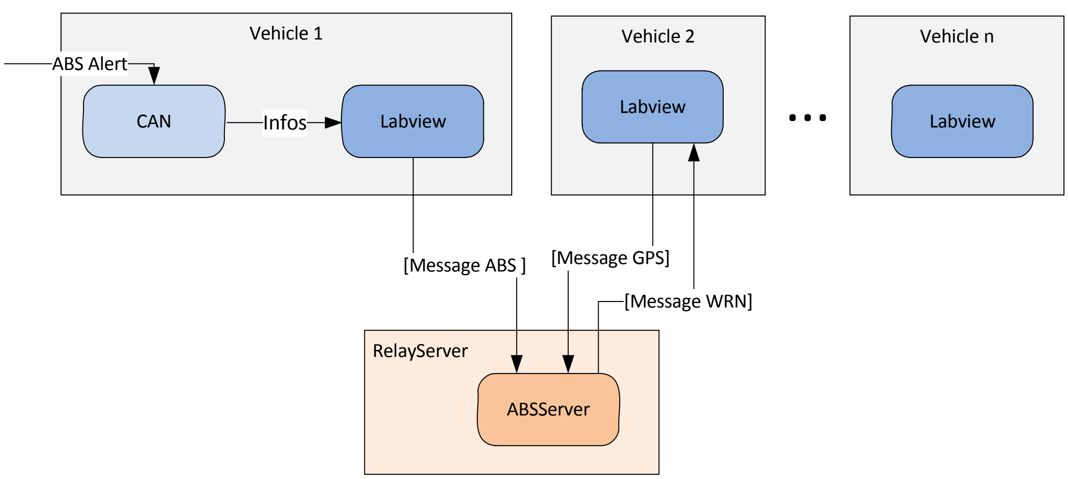

Figure 11 Diagram for the ABS warning activation by V2I

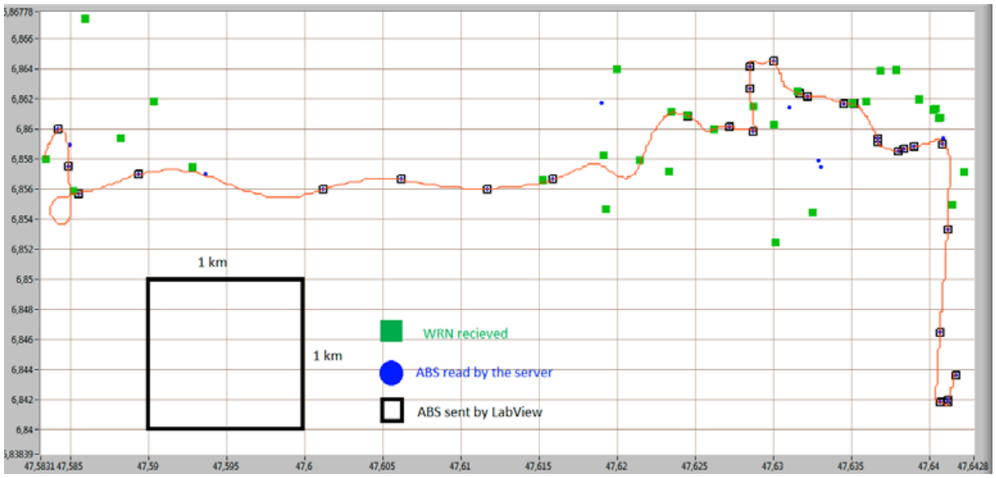

Figure 12 Comparison between data from Labview and the server

6 Conclusions and future work

The CAN is widely used in the light-duty automotive and industrial market systems such as motor vehicles, utility vehicles, and industrial automation. It provides a lot of information about the vehicle, which are collected from different sensors. But in the automotive industry, every automobile company has its own CAN bus system; they are private and closed bus, and the frame identifiers of the CAN bus system are top-secret. Therefore, no communication with the existing CAN bus can be easily established. In this paper, an embedded system was introduced to extract and record information about in-vehicle CAN bus events. The system is interconnected to the vehicle existing CAN system and decodes information messages. The system uses also wireless communication modules, namely ZigBee and GPS/GPRS, to share relevant information. An experimental test was conducted and preliminary results are reported and show that extracting this information are relevant to develop safety applications. Additional experiments together with extracting more information are part of our ongoing work.

Several applications could be implemented on top of the embedded system for sharing in-vehicle information between vehicles. As described above, in-vehicle ABS activation might indicate bad road condition and if this information is shared between vehicles, other drivers can be warned to take preventive actions before getting to a dangerous situation. If a driver perceives an object on the road, he/she will be able, via the touch screen, to indicate this information and it could be also shared to warn the drivers of approaching vehicles. For example, when the motor oil indicator lamp is on, a sudden break might happen, the driver should be warned and other surrounding vehicles will be informed as well, to avoid this potential obstacle. A sudden drop of a vehicle speed, if shared with following vehicle drivers, could help adapting the speed and avoiding rear collisions.

Acknowledgements

This work is supported by the EU project ASSET (Advanced Safety and Driver Support for Essential Road Transport, 2008-2011).

References

1. Guo S.: 'The Application of CAN-bus Technology in the Vehicle', 2011 International Conference on Mechatronic Science, Electric Engineering and Computer, pp. 755- 758, August 19-22, 2011, Jilin, China
2. Sathyanarayana, A., Boyraz, P., Purohit, Z., Lubag, R., Hansen, John H.L.: 'Driver Adaptive and Context Aware Active Safety Systems using CAN-bus signals', IEEE Intelligent Vehicles Symposium, pp. 1236-1241, University of California, San Diego, CA, USA, June 21-24, 2010
3. Dar, K., Bakhouya, M., Gaber, J., and Wack, M.: 'Towards a Decentralized Architecture for Information Dissemination in Inter-vehicles Networks'. In AICCSA proceeding, 2010.
4. Dar, K., Bakhouya, M., Gaber, J., Wack, W., and Lorenz, P.: 'Wireless Communication Technologies for its Applications'. In IEEE Communication Magazine, 2010, volume 48, pp. 156-162.
5. Ait-Cheik-Bihi, W., Bakhouya, M., Nait-Sidi-Moh, A., Gaber, J., Wack, M.: 'A Platform for Interactive Location-Based Services'. In 8th International Conference on Mobile Web Information Systems, Procedia Computer Science, 2011, Volume 5, pp.697-704 Elsevier.
6. Fuchs, S., Rass, S., and Lamprecht, B.: 'Context-Awareness and Collaborative Driving for Intelligent Vehicles and Smart Roads'. In GIIS proceedings, 2007.
7. National Instruments Corporation, CompactRIO and LabVIEW Development Fundamentals Course Manual, November 2007 Edition, Part Number 324516B-01, 2007.
8. Zhou Y., Wang, X. and Zhou, M.: 'The Research and Realization for Passenger Car CAN Bus'. In the 1st International Forum on Strategic Technology. 18-20 Oct. 2006, pp. 244-247.
9. Cai, S., Becherif, M., Bakhouya, M., and Wack, M.: 'A Context Aware Embedded System for Intelligent Vehicles and Smart Roads'. In: ACM ICPS, UPC Workshop, 2009.
10. Song, D., Li, J., Ma, Z., Li, Y., Zhao, J., and Liu, W.: 'Application of CAN in Vehicle Traction Control System'. In IEEE International Conference on Vehicular Electronics and Safety, 2005, pp. 188-192.
11. Johansson, K. H., T¨orngren, M., and Nielsen, L.: 'Vehicle Applications of controller area network. In Handbook of Networked and Embedded Control Systems', D. Hristu-Varsakelis and W. S. Levine Eds. Birkh¨auser, 2005, pp. 741–766.
12. Marino, A., and Schmalzel, J.: 'Controller Area Network for in-Vehicle Law Enforcement Applications'. In IEEE SAS'07 Sensors Applications Symposium, pp. 1-5, 2007.
13. Kim, S., Choi, J., Kwak, D. , Angkititrakul, P., Hansen, J.H.L.: 'Analysis and Classification of Driver Behaviour using In-Vehicle CAN-Bus Information', Biennial on DSP for In-Vehicle & Mobile System, Istanbul, Turkey, June 2007.
14. Cai, S., Becherif, M., Henni, A., Wack, M., and Aboubou, A.: 'Processing of Information Gathered from Hybrid or Electrical Vehicle for Reduction of Fuel Consumption and Gas Emission', The Mediterranean Journal of Measurement and Control (MedJMC), ISSN : 1743-9310, Vol. 7, Issue 2, pp :236-242, April 2011.