EVS26
Los Angeles, California, May 6 - 9, 2012

Web management of electric vehicle fleets

Franco Callegati, Alessio Pilati, Marco Ramilli, Claudio Rossi
University of Bologna D.E.I. "G. Marconi"
{franco.callegati, alessio.pilati, marco.ramilli, claudio.rossi}@unibo.it

Abstract

An architecture for enabling the internet connection of a fleet of electric vehicles is presented in the paper. The connection is based on the on-board integration of an Arduino platform with the CAN-bus communication network of the vehicle powertrain. The system allows a plurality of networking interface: GPRS, LAN, Wi-fi for the present and Wi-Max, LTE for the future. The proposed system manages the main powertrain parameters and warning/fault condition. Local and remote data storage is possible, such as direct web interface of the vehicle powertrain. Data post processing from the remote server allows the implementation of an automatic diagnostic procedure for the detection of incipient faults in the powertrain.

Keywords: Battery Electric Vehicles (BEV), communication, data acquisition, powertrain

1 Introduction

It is a common opinion that electric vehicles are now ready for mass series production. The increased range and performance of electric cars and the expected evolution in the near future [1] is now compatible with a wider range of use cases, in particular in the urban environment, and the number of potential customers is growing rapidly.
The two main obstacles to the diffusion of these new means of transportation are given by the initial high cost and by a diffused suspicious towards the reliability of this new technology. Many strategies have been introduced for cutting the initial price of an electric vehicle, for example throughout the introduction of new battery technologies [2] and of new financial management (e.g the leasing of the battery pack).
Nonetheless a lot has still to be done to show the final user that this is a mature technology that can effectively compete with the traditional vehicles. Several are the issues to be addressed, mainly related to the new power source and engine. Will the electrical engine be effective in real life driving conditions? Is there a chance that the battery main drain out of power unexpectedly and leave the user stuck in the traffic? Is there a chance that when the user needs the vehicle it is still stuck re-charging? These are just trivial examples of questions which must be fully answered before electrical city cars may start replacing conventional cars in real life.

In these paper we argue that a solid contribution to answering these question may come by developing a strong interaction between the electrical vehicle and the Internet. Vehicular networks usually called VANET are a well known topic in networking [3], but the work done in this field to date has been mainly related to the analysis of transmission issue and the design of protocols for vehicle to vehicle or vehicle to ground communication. The interaction with the vehicle control is a topic less analyzed, in particular for what regards the interaction with the vehicle control system. Communication standards are also under study to provide communication between the vehicle and the energy provider network during the re-charging phase [4]. Here the focus is mainly on "operational" purposes, such

as authentication, billing etc., strictly related to the re-charging phase. In all these cases a limited amount of attention is devoted to applications which could exploit the communication with the vehicle control system to enhance functions related to the vehicle final user.
In this paper we describe a VANET solution oriented to the vehicle to ground communication, which may easily escort the new drivers in using the electric technology, exploit social networking applications to share personal experiences in favor of the new technology and allow the manufacturer (or its service branches) to perform a massive control over the circulating vehicles, with reference to the mode of use and state of condition of the vehicles.
The proposed solution is a rather simple and cheap communication system, based on state of the art micro controller technology [5], connecting directly the powertrains of a fleet of electric vehicles with the web and remote data centers, where data are analyzed and decisions are taken. Basically the system is made by a "local device" integrated with the local high speed CAN bus of the powertrain. A set of information representing the state of the powertrain are collected by the "local device", pre-processed and then transmitted to the "remote server" over the Internet by opportunistically using the most viable networking technology available. In the experiment here presented GPRS/UMTS mobile networking is used when the vehicle is moving, while high speed wired Ethernet or Wi-Fi is used when the vehicle is re-charging at a re-charging station or at home.
The paper is organized as follows. In section 2 an overview of the considered power system is given, with particular reference to the CAN bus characteristics and relevant messages. In section 3 the networking module is described. In section 4 is presented an overview of the applications which could be implemented exploiting the proposed system. In section 5 the performed experiments is reported with related results. Finally is section 6 conclusions are drawn.

2 The LEMAD power system

The powertrain used in this work is designed for full electric vehicles and was developed by the Laboratory of power electronics, electric drives an electric machines (LEMAD) at the University of Bologna. This powertrain is designed for the propulsion of lightweight vehicles, below 1 ton of maximum mass, i.e. city cars and leisure vehicles as side-by-side.
The LEMAD powertrain is an integrated system that can be described by using two schemes, one for the power stage which is shown in Fig. 6 and one for the control and communication system which is shown in Fig. 2.

The power circuit is a traditional full-electric driveline. A battery pack, composed by n-cells in series is connected through the main DC bus to the traction inverter. This inverter supplies the traction motor, which is usually integrated with a mechanical reduction gearbox and differential.
The main DC bus supplies a plurality of other power converter modules: the DC/DC converter which provides power to the PWT auxiliary circuits, the DC/DC converter which provides power to the vehicle loads, the inverter for the motor of the air conditioning compressor, the on-board battery charger.
The battery pack is conditioned by using a modular Battery Management System (BMS). Each module is able to monitoring temperature and voltage of 8 cells. The BMS modules implement the passive equalization of cells during recharging by dissipating a small portion of the recharge current.

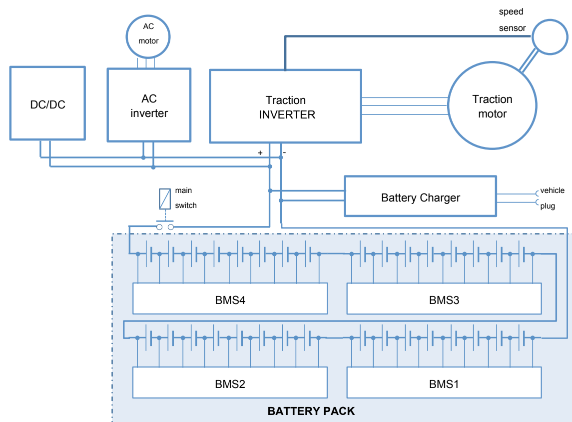


Figure 1: Schematic power circuit of the full electric powertrain

Each device is controlled by a microcontroller and all devices are interconnected through a high speed CAN bus network. On each device, the control system architecture is based on a multi-layer structure. The control algorithm is split in low level, high level and communication level. In particular the high level control algorithm is interconnected through the CAN bus with other peripherals for data sharing, while the low level take the references from the high level and controls the hardware device (power electronics, signal conditioning, etc..).
Fig. 3 shows the main functions of the Traction Control Module which are implemented on the traction inverter control board. Briefly, in this control module, the high level control functions collects the data from the Vehicle Interface and the BMS master and generates the references for the motor control algorithm assigned at the low level. The high level hosts:

- the functions related to the vehicle drivability such as pedal ramping, speed regulator,

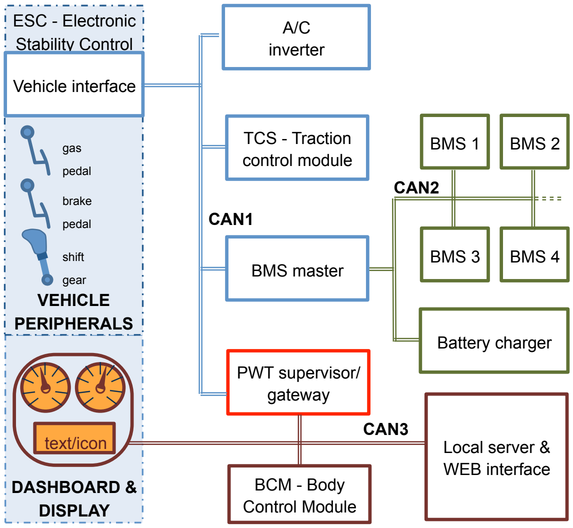


Figure 2: Interconnection of devices by the CAN bus

reference smoother;

- the torque limits map for the different driving mode and vehicle operation;
- the power limiting map for battery overvoltage and undervoltage protection and for overheating;
- all protection and safety tasks directly related to the inverter operation.

The low level receives the torque reference from the high level and implements the motor control algorithm driving directly the inverter commutation. This level hosts:

- the I/O management;
- the flux-speed estimator;
- the field oriented torque control;
- the flux weakening strategy;
- the current regulator and the space vector modulator.

The CAN-bus network enables the communication among the control modules of the PWT. It is based on three electrically separated network called CAN1, CAN2, CAN3 as it is shown in Fig. 2.

CAN1 is reserved for communication among power devices of the powertrain: traction control module, vehicle interface, master BMS and air conditioning inverter.

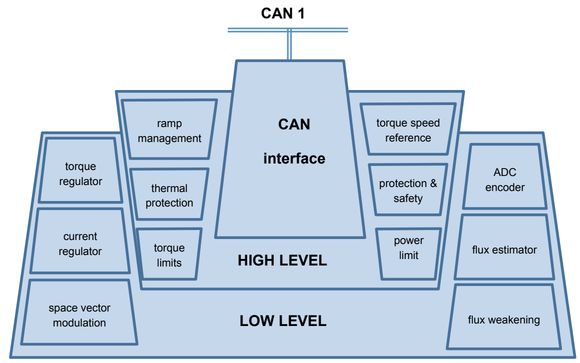


Figure 3: Main functional block assigned at the two level of the Traction Control Module

CAN2 is reserved for communication among devices of the battery systems: BMS master, BMS modules and battery charger. The BMS master is than connected with both CAN2 and CAN 1 for enabling the power limiting function to the powertrain due to the battery status.

CAN3 is reserved for communication with monitoring and external devices, such as the car dashboard, the car body control module, the on board data storage and the car webserver.

The same CAN protocol (same library and semantics), the same watchdog and the same coherence checking is implemented on the three CAN networks.

The module called "PWT supervisor and gateway" patrols the connected units (on CAN1) for a staying alive check of the main PWT control modules and operates as gateway between the PWT units and external peripherals. The "gateway" selects a list of PWT variables available on the devices connected on CAN 1 and supply this data on CAN 3 by using CAN-open format. The data selected for transmission to CAN 3 are classified by priority and transmission rate, as shown in Tab. 1. Thirteen priority level and three timestamp are used. More than one messages can be assigned at the same priority level. The priority level is encoded directly in a portion of the CAN message ID.

On CAN 3 are broadcasted both numeric data, and status condition. The list of numeric data with their priority level is given in Tab. 2. Status condition includes the transmission of the active operating mode of the PWT and of the warning/fault condition. The PWT status is sent every 10 ms with high priority and the most important status information are:

- Operating mode
- Selected gear and operating gear
- Driver error in gear/pedal sequence

Scansion Rate [ms]	PRIORITY	ID_type
10	VERY HIGH	1
10	HIGH	2
10	HIGH	3
10	HIGH	4
10	HIGH	5
100	MID	6
100	MID	7
100	MID	8
100	MID	9
1000	LOW	10
1000	LOW	11
1000	LOW	12
1000	LOW	13

Table 1: Scansion rate, priority and ID on CAN 3

ID	Variable	Unit	Rate [ms]
3	MOTOR CURRENT	[ARMS]	10
3	BATTERY CURRENT	[A]	10
6	MOTOR WINDING TEMPERATURE	[^{o}C]	1000
3	DC BUS VOLTAGE	[V]	10
6	CHARGE LEVEL	[%]	100
7	CHARGE CURRENT	[A]	1000
7	CHARGED Ah	[Ah]	1000
7	N. of EQUALIZED CELLS		1000
5	MOTOR SPEED	[rpm]	10
6	INVERTER TEMPERATURE	[^{o}C]	100
6	AUX. BATTERY VOLTAGE	[V]	100
6	MAX. BATTERY TEMPERATURE	[^{o}C]	100
4	LOWER CELL VOLTAGE	[V]	10
4	HIGHER CELL VOLTAGE	[V]	10

Table 2: List of numeric data broadcasted on CAN 3 with related transmission rate.

Any warning/fault condition, if active, is sent with the higher priority every 10 ms. About 80 warning/fault condition for each control module can be managed and sent on CAN 3.

3 Enabling the vehicle Internet connection

The connection with the Internet of the vehicle is implemented by means of an embedded system providing networking functions with the cost, complexity and reliability profile of a microcontroller system for industrial applications. It was developed in the networking laboratory DEILab at the University of Bologna.
The system is based on the Arduino hardware and software platform [5], which has the interesting feature of being an open project bringing to the final user a set of low cost and widely available modules (called shields) implementing a large variety of functions. This modular approach allows for a rather fast implementation of systems providing a series of different functions.
In this project the Arduino UNO [6] mother board was used as the basic building block. This board is equipped with an ATmega328 microcontroller, which is equipped with a 32 Kbyte Flash Memory and has a clock speed of 16 MHz. The Arduino UNO is also equipped with the USB interface needed for programming, which is also used as a general purpose serial interface. The board is shown in Fig. 4.

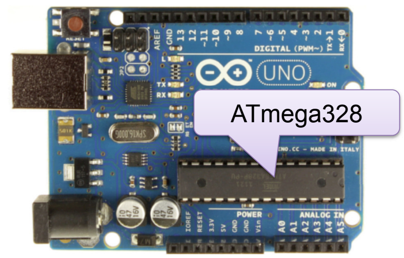


Figure 4: The Arduino UNO board

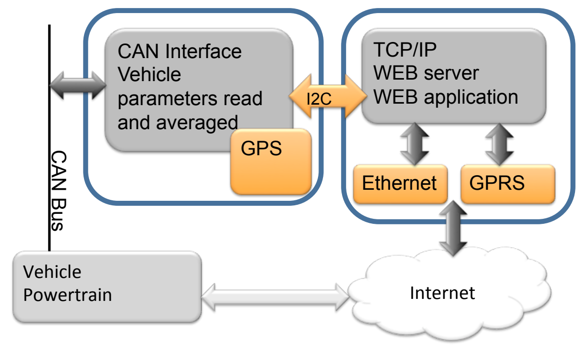


Figure 5: Block diagram architecture of the vehicle networking system.

The overall block diagram of the communication system is shown in Fig. 5. The described system is made of two major sub-blocks. The former is responsible of the communication with the vehicle management system, the latter is responsible of implementing the interface with the network. The latter block could be made of more than one board with suitable network interface, to make available different networking technologies at once. Data are exchanged between boards using an I2C serial bus [7].
This modular approach is very effective in our view. It makes available several networking opportunity at once in a user selectable way. In fact a user may decide to equip the vehicle with just one or with all the possible interfaces by choosing which of the Ethernet, Wi-Fi and GPRS modules to take on board. In case several interfaces are available the system may opportunistically use the one considered optimal depending on its current conditions. Typically wireless technologies able to guarantee a wide area coverage (GPRS/UMTS) are more suitable to provide connectivity when the vehicle is in motion, while wired or wireless technologies (Ethernet, Wi-fi) providing larger band but shorter reach are more suitable to provide connectivity when the vehicle is standing still parked and/or re-charging.
Moreover this architecture leaves open the door

to any possible future development of the vehicle networking features. The choice of using Internet standard protocols to present the information to the final user will immediately exploit new features that may arise in this field, while the modular hardware architecture may easily accommodate new network interface that may become available by adding modules that are equivalent to these ones in term of mother board and related network operating system, but will be equipped with the new interfaces (for instance the wide range wireless technologies such as Wi-Max or the newer LTE).
The system is also equipped with a large capacity storage unit (SD card). The relevant vehicle data may be stored in the storage unit for backup reasons or as a sort of buffer for periods when the connections are lost. In the latter case the stored data may be uploaded to the server by the web application when the connection is re-established.
Overall this rather simple system fulfills a number of tasks:

- read relevant operating data from the vehicle power train;
- parse these data and covert them to an "Internet friendly" format;
- make the data available for consultation to the final users in a web format, meaning as a set of web pages which can be navigated with a conventional Internet browser;
- write the data on a remote server by means of an suitable Internet application so that this data may be processed by an off-line application;
- receive feedback messages from the server and more generally from the Internet which can be displayed to the driver of the vehicle for information or alarm purposes.

3.1 CAN Interface

This is the front end to the vehicle control system. The mother board is equipped with a shield implementing the interface with the CAN bus. The module reads the messages made available by powertrain, and elaborates them for monitoring purposes. Typically the values are averaged over a period of a few seconds and a history of about a minute is kept in the local memory.
The board is also equipped with a GPS modules that allows tagging the values stored in memory with time and location stamps.
The measured values, plus time and location, are periodically passed to the other modules.

3.2 Local web server

This module enables the local web server interface. It is equipped with a high speed networking interface, which may be either Ethernet for wired connectivity or Wi-fi for wireless, and with a networking operating system which implements the whole TCP/IP stack plus including the HTTP application protocol and acts a full functional web server. By means of this module the vehicle may connect to a high-speed local area network (LAN), typically when not in motion.
When the board in connected to the LAN, the web pages of the local web server may be browsed by any user over the Internet by using a conventional web browser. For security reasons a proper authentication may obviously be required. The pages reports a graphical synthesis of the past vehicle behavior. The kind of information, the amount of time reported etc. can be configured by the user as well as the graphical look of the web pages by means of information stored into the standard SD card which is on board.
The web pages implemented for this local web server have been explicitly written to minimize the data transfer load and maximize the effectiveness of the user experience. Therefore they implement state of the art web technologies such as Ajax to enable on the fly refresh of the displayed values with limited transfer of data.

3.3 Internet based remote data storage

This a board is much alike the previous one but equipped with a geographical low speed GPRS network interface. It gets the same data provided by the CAN interface but, instead of making them available as browsable web pages, periodically upload them on a remote database that stores the whole history of the vehicle behavior.
This is done exploiting the GPRS network connectivity that is available also in motion and, again, standard Internet protocols. The board is equipped with the same network operating system as the previous one but, instead of implementing the local web server, upload the information to the remote database by means of periodic HTTP POST messages. The data stored in the database may be used for any sort of off line vehicle monitoring by authorized users.

3.4 Inter-board Communication

The Arduino based mother boards which makes the Internet interface of the vehicle communicate one another using an I2C serial bus which allows a very limited and compact software. The bus is implemented on two analog pins of the ATmega328 microcontroller. The speed of the I2C bus is one of the possible bottlenecks of this

system. Therefore the amount of data to be exchanged over the bus must be carefully dimensioned.
In our implementation this was done by buffering and pre-processing the data flowing from the power train on the CAN bus in the Arduino UNO board that implements the CAN interface. The vehicle relevant data are summarized and groomed with the aim to keep the overall throughput towards the boards interfaced with the Internet below the limits of the I2C bus capacity.
More quantitative details about this issue will be given in Sec. 5.

4 Application Scenarios

Given the vehicle is equipped with the Internet interface which are the application scenarios that can be envisaged? Again many VANET application have been considered already, mainly focusing on traffic control and entertainment. In this work, as already discussed int he introduction, the focus is more on the enhancement of the vehicle functions, in conjunction with the specific features of the electrical vehicle. Indeed one major concern is to guarantee the full safety, security and privacy of the car end-user. Therefore the application scenarios of this new technical feature must be carefully designed and analyzed. A wrong user perception of the services offered by the Internet interface would indeed turn this feature into a con rather than into a pro.
In this section we try to outline a few of these application scenarios which we argue could be acceptable for the final user.

4.1 Recharging control

The recharging process is significantly different in electrical vehicles when compared to traditional thermal engine vehicles, mainly because of the duration of the recharging phase (typically longer a conventional refueling) and of the availability of the recharging station. In both cases the availability of the Internet interface may be of help in making the overall recharging process more effective.
The most trivial application is that to allow the final users to monitor the power level of the vehicle remotely. A typical scenario is that of a user leaving the vehicle recharging at a recharging station nearby his working site. By remotely checking the re-charging level via the web, for instance by using the browser in his office computer or a simple mobile application running on a smartphone, the user may be aware of when the recharging is complete and can then move the vehicle from the recharging station to his office parking lot. Similarly the Internet interface of the vehicle may send the user a text message or an e-mail when the recharging is complete. This will allow a better use of the re-charging stations, as well as better perception of the re-charging phase by the user.
But indeed more complex functions could be implemented. For instance consider a scenario where several recharging stations of different providers are available in a given area, the final user could be advised through a text message about which are the free stations or the cheapest ones.

4.2 Messaging

The opportunity to exchange text messages with the vehicle with standard and widespread technologies such as SMS or e-mail opens the field to a great variety of applications. A rather trivial example is the aforementioned one about recharging level advice, but many others could be imagined from traffic control, to advertisement, to social networking.
For instance the final user could use this feature to send to the vehicle a message which will remind him of a duty to be performed as soon as the user starts moving with the vehicle (we can call this a “remember to buy the milk” application).
The very interesting feature of this approach is that the ”design” of the application is left to the user, since the general purpose technology is fully transparent and very familiar to wide public. The history of the Internet already proved that this approach is in general very successful in fostering those networking phenomena that drive the success of new applications and systems.

4.3 Maintenance and fleet control

The data stored in the remote web application are post-processed by an electric fleet manager application in order to detect anomalies, warning conditions or faults in the vehicle. The output of this application towards the driver are given by different levels of driver alerting or by calling emergency support in case of a serious fault impairing the motion of the vehicle.
The output of the application toward the vehicle maintenance service is mainly directed to the recognition of incipient faults in the powertrain components (power electronics, machine, batteries) and to the monitoring of the performance decrease in the cells of the battery pack.
The data from the whole fleet, once aggregated, can be used for statistical analysis of the vehicle power system and battery behavior, to enable improvements and/or detection of widespread anomalies.
In all these applications it is important to guarantee the privacy of the vehicle final user. Therefore the data must be properly sanitized [8, 9] to

avoid that a correlation between vehicle owner, GPS tracks and vehicle data performance may be maliciously used to backtrack the final user behavior and habit. This is a topic well known from similar application in other sectors and viable solutions are already available.

4.4 Social Vehicles

Social networks such as FaceBook (www.facebook.com), Twitter (www.twitter.com), Google+ (plus.google.com) and many others have recently become part of our daily life. People likes to share what they are doing and where they are going, by modifying their status on "public" profiles often called "Time Lines".
The Internet interface may be used to enable a sort of "social interface" which let the car directly sharing on public time lines information such as the current position of he vehicle, what is the temperature of the vehicle, what is the environment temperature, what is the state of the battery and so on.
The car could upload on social networks even its own statistics on mileage, power consumption and speed, thus fostering a sort of "behavioral competition" between final users for better use of energy etc..

4.5 Environment Monitoring

Finally the Internet interface here described, being based on a micro controller hardware platform is very keen in interfacing with sensors. Besides temperature metering already available in most vehicles, cheap sensors measuring humidity, pollution and/or other environmental parameters could be integrated in the system. The monitored valued may easily be uploaded to the web application for further use and processing.
In particular by constantly sending information on social networks and/or on external data base it is possible to monitor pollution in given areas with a high degree of detail, to capture micro-pullution phenomena.

5 Experimental Set-up

For demonstrating the capability of the proposed web interface system for electric vehicle, a board prototype containing all the function described in Section 3 has been realized and connected at the CAN 3 of the powertrain.
The experimental powertrain is based on the configuration described in Section 2. It is designed for supplying an induction motor with an output power rating of 10 kW, by using a battery pack composed of 32 ion-lithium cells connected in series. Tab. 3 contains the main data of the powertrain. Fig. 6 shows the main box containing the traction inverter, vehicle interface and DC/DC converters.

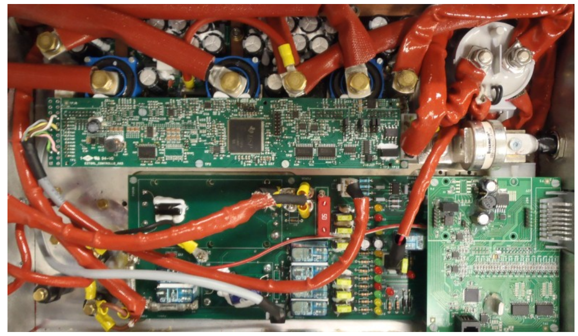


Figure 6: Main box of the experimental powertrain.

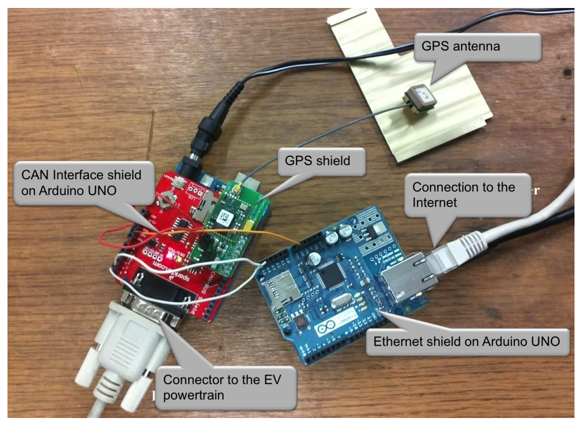


Figure 7: The web interface implemented with off-the-shelf pre-mounted Arduino boards and shields.

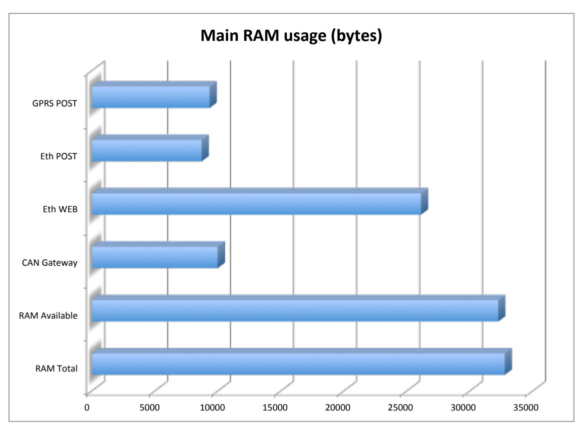


Figure 8: ATMega328 system RAM usage on the Arduino UNO board for the various system options.

Traction drive - power module	
rated supply voltage	105 [V]
min supply voltage	80 [V]
max. supply voltage	135 [V]
max. n. of lithium cells in series	32
rated output current (continous duty)	150 [ARMS]
maximum output current (60 s)	400 [ARMS]
rated output power	18 [kVA]
maximum output power (60 s)	42 [kVA]
Traction drive - motor control algorithm	
basic type	general rotor flux field oriented control
motor type	IM: induction machine SPM-SM: surface permanent magnet synchronous machine ARPM-SM: assisted reluctance permanent magnet synchronous machine R-SM: reluctance synchronous machine WR-SM: wound rotor synchronous machine
field weakening	maximum output power over a wide speed range
modulation strategy	SVM: Space Vector Modulation
maximum output fundamental frequency	400
switching frequency	8
Traction control module	
torque reference	mixed direct torque and speed control
torque maps	four indipendentlimitation group over full quadrant
power limitation	direct input power limitation for battery single cell overvoltage-undervoltage protection
gas and brake pedals management	ramps and smoothing for both direct torque and speed control
temperature limitation	proportional reduction of output currrent for overtemperature in motor and drive
Battery managment system	
configuration	master-slave
battery cells for slave module	8
temperature input for slave module	2
cell voltage resolution	5 [mV]
cell voltage refresh rate at master	9 [ms]
equalization method	passive
equalization current	0.6 [A]
battery charger control	reference charge current
SOC estimation	based on cell model. Computation with OCV feedback correction
protection	individual cell overvoltage-undervoltage
Auxiliary systems	
DC/DC conv. for 12V vehicle power system	400 [W]
12 V circuits	power-on, power-off, protection, power distribution
additional controls	passenger compartment heating ventilation air conditioning (HVAC)

Table 3: Experimental powertrain specification.

The power train was equipped with the web interface. The picture in Fig. 7 portraits the web interface built with off-the-shelf electronics in a ab set-up. It is made of the Arduino UNO board with CAN interface on the left, connected via the I2C bus to the Arduino UNO board equipped with web server and Ethernet interface on the right.

The experiment was used to prove the overall feasibility of the approach chosen. First of all it was important to test that all the required functions could be implemented on the hardware platform chosen. In Fig. 8 is plotted the ATMega328 programmable RAM occupancy of the various software modules. Overall the RAM available is limited to 32256 bytes, given by the total RAM on board minus the space required by the boot loader necessary to program the microcontrollers. The module implementing the CAN interface and the web applications downloading the monitored data to a remote server via Ethernet or GPRS are indeed well below the limit and use approximately 50% of the overall capacity. The module implementing both the web application and a full functional web server with web pages which can be navigated from a remote browser for on-line monitoring of the state of the vehicle is more complex and requires 26 of the 32Kbytes available, for an overall RAM occupancy of approximately 80%, still acceptable for normal operating conditions.

Another important issue to test was the compatibility of the overall system processing capacity with the workload related to the amount of incoming messages over the CAN bus. Sending data on the web at the speed they flow on the CAN bus for real time control of the vehicle is not a viable option. In the implementation the Arduino board implementing the CAN interface acts as follows:

- reads all the CAN messages;
- averages the parameters read over a subset of M values;
- stores the average in an array in its RAM (arrays are different for different ID-nodes, so that an array stores consistent data values);
- the arrays have a fixed dimension of N and when are full a packet is built and sent to the Arduino board implementing the Internet interface over the I2C bus.

In the experiment here presented $N = M = 20$, meaning that the parameters related to a given vehicle function are averaged over a set of 20 values and then submitted in series of 20. Therefore the board implementing the Internet interface receives packets carrying the summary of 400 values related to a given parameter. In our view this solution realizes a reasonable trade-off between accuracy of the data sent on the Internet. We logged 30 minutes of traffic flowing on the I2C bus while the system was operational capturing approximately 700 packets of data. The average inter-arrival time was 2.7 sec. and an average packet size 150 bytes. This resulted in a load on the I2C bus of approximately 450 bit/sec. The choice made guarantees a load on the I2C bust well below it nominal capacity and does not cause any buffer overload at the receiver.

Finally it is important to mention that the powertrain, equipped with the vehicle web interface system, is installed on two full-electric vehicles manufactured in Italy: the four seats city-car *BELUMBURY DANY* shown in Fig. 10 and the side-by-side *TGS PIRATE* shown in Fig. 11.

Figure 10: Picture of the electric four seat city car Belumbury Dany equipped with the web interface system

.

Figure 11: Picture of the electric two-seat buggy TGS pirate equipped with the web interface system.

6 Conclusions

In this paper we have described the implementation of the Internet interface for an electrical vehicle. The network interface is built using a micro controller hardware platform which is interfaced directly with the CAN bus carrying the information regarding the vehicle powertrain behavior.

In the paper we have discussed the architecture and the performance issues of the system proposed, showing it is compatible with the current vehicle technology.

References

[1] A.Y. Saber, G.K. Venayagamoorthy, *One million plug-in electric vehicles on the road by 2015*, Proc. of 12th International IEEE Conference on Intelligent Transportation Systems, 2009. ITSC '09. Oct. 4-7 2009. St. Louis, MO (USA). ISBN: 978-1-4244-5519-5

[2] A. Burke, *The development and performance of Lithium batteries of different chemistries and comparison with other advanced batteries*, Proc. of EVS 24 Electric Vehicle Symposium, May 13-16 2009. Stavanger, Norway.

[3] K. L. Hannes Hartenstein, Ed., *VANET Vehicular Applications and Inter- Networking Technologies*. Wiley, 2010.

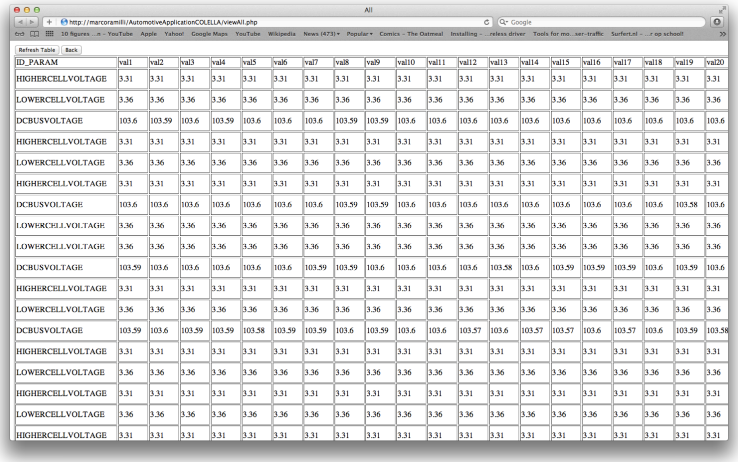
[4] *Road vehicles Vehicle to grid communication interface Part 1: General information and use-case definition*, Draft International Standard ISO/DIS 15118-1, 2012.

[5] *Arduino*, `http://www.arduino.org`, accessed on 2012-01-06.

[6] *Arduino UNO*, `http://arduino.cc/en/Main/ArduinoBoardUno`, accessed on 2012-01-06.

[7] *I2C Bus*, http://www.i2c-bus.org, accessed on 2012-01-06

[8] A. Amiri, *Dare to share: Protecting sensitive knowledge with data sanitization*, Journal of Decis. Support Syst., February, 2007.



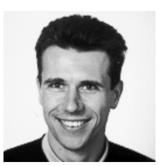
ID_PARAM	val1	val2	val3	val4	val5	val6	val7	val8	val9	val10	val11	val12	val13	val14	val15	val16	val17	val18	val19	val20
HIGHERCELLVOLTAGE	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31
LOWERCELLVOLTAGE	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36
DCBUSVOLTAGE	103.6	103.59	103.6	103.59	103.6	103.6	103.6	103.59	103.59	103.6	103.6	103.6	103.6	103.6	103.6	103.6	103.6	103.6	103.6	103.6
HIGHERCELLVOLTAGE	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31
LOWERCELLVOLTAGE	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36
HIGHERCELLVOLTAGE	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31
DCBUSVOLTAGE	103.6	103.6	103.6	103.6	103.6	103.6	103.6	103.59	103.59	103.6	103.6	103.6	103.6	103.6	103.6	103.6	103.6	103.6	103.58	103.6
LOWERCELLVOLTAGE	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36
LOWERCELLVOLTAGE	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36
DCBUSVOLTAGE	103.59	103.6	103.6	103.6	103.6	103.6	103.59	103.59	103.6	103.6	103.6	103.6	103.58	103.6	103.59	103.59	103.59	103.6	103.59	103.6
HIGHERCELLVOLTAGE	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31
LOWERCELLVOLTAGE	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36
DCBUSVOLTAGE	103.59	103.6	103.59	103.59	103.58	103.59	103.59	103.6	103.59	103.6	103.6	103.57	103.6	103.57	103.57	103.6	103.57	103.6	103.59	103.58
HIGHERCELLVOLTAGE	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31
LOWERCELLVOLTAGE	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36
HIGHERCELLVOLTAGE	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31
LOWERCELLVOLTAGE	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36	3.36
HIGHERCELLVOLTAGE	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31

Figure 9: A simple example of a web page reporting the vehicle operating parameters, as stored on a remote web server by the web application running on the Internet interface

.

[9] S.R.M. Oliveira, O.R. Zaiane, *Protecting sensitive knowledge by data sanitization*, Third IEEE International Conference on Data Mining (ICDM) , 19-22 Nov. 2003.

Authors



Franco Callegati is Associate Professor of Telecommunication Networks. He has an expertise in broadband network protocols and architectures, optical networking and network security. He has been active in EU funded projects since 1995. He contributed to several books and published more then 100 papers in international journal and conferences. He is Senior member of the IEEE.

Alessio Pilati received the Ph.D. degree at the Department of Electrical Engineering at the University of Bologna in 2012. He is Post Doctoral Researcher with the Electrical Machines, Drives and Power Electronics. His present research activity is devoted to the development of advanced digital control systems for electric machines.

Marco Ramilli is Post Doctoral Researcher. He has an expertise in computer security, malware and penetration testing methodologies. He has been working with US Government and International Universities on Electronic Voting Machine penetration testing. He is a member of the IEEE.

Claudio Rossi is Assistant Professor of Electrical Machines, Drives and Power Electronics. His present research activity is devoted to power electronics and drives for electric vehicles and renewable energy systems. He is a Registered Professional Engineer in Italy.