

Understanding future customer needs by monitoring EV-drivers' behavior

Marina de Queiroz Tavares¹, Jonas Gartmann, Flavio Perucchi, Franz Baumgartner² and Maria Youssefzadeh³

¹Center for Signal Processing and Communications Engineering, Zürich University of Applied Sciences,

²Institute of Energy Systems and Fluid Engineering, Zürich University of Applied Sciences,

³Center for Aviation and Transport Systems, Zürich University of Applied Sciences,
8401 Winterthur, Switzerland

E-mail: ¹dqtm@zhaw.ch, ²bauf@zhaw.ch, ³youn@zhaw.ch

Abstract—Future development of electrical vehicles and related services need to incorporate knowledge of the customer habits and the detailed battery behavior such as loading and consumption profiles. For example the planning of the appropriate charging infrastructure (location and number of stations) needs a lot more data about the real usage of EV's. Collecting this data, and making it remotely available is the function implemented in the EV-Monitor and GPS-Tracker. The EV-Monitor is a microcontroller based data logging system specialized in measuring and recording the vehicle charging current, position (via GPS coordinates) and other key data available over the CAN bus of the vehicle. The data is periodically sent via wireless connection (GSM/GPRS) to a network server, and can be visualized with a platform independent graphical interface (JAVA based). The solution implemented is flexible, configurable (firmware and software self-coded, and hardware allowing connection of several additional analog and digital peripherals), easy to install and price attractive. The first prototype of the EV Monitor was tested successfully in summer 2010. Currently a 2nd prototype series is being produced for tests with the local energy supplier of the greater Zurich area (EKZ). *ZHAW Copyright.*

Keywords— Electrical Vehicles, Battery Behaviour, Customer Habits.

1. Introduction

The limited resources of fossil fuels as well as the rising greenhouse gas emissions of transports pose a global challenge. Part of the solution is seen in switching to zero emission electric vehicles. However there are still several open issues to be considered for the expansion of the EV's market.

First the acceptance of electric vehicles amongst average car buyers is still very low. In view of the substantial investment and the long-term commitment involved in buying a new car, the characteristics of electric cars are still not well known enough by the public.

Second car manufacturers and infrastructure providers do not know enough about the specific behavior of electric car drivers, so that they could adjust their product to the market.

Third there is a lack of data that would allow electricity providers to estimate the consumption of electric cars, the appropriate locations for public and private filling stations, and the strategies to balance the electricity demand and eventually integrate vehicle's batteries in smart grid systems.

Therefore it was decided in this project to develop a data logging device to analyze the interaction of EVs with the charging utility grid. By equipping test fleets of electrical vehicles with the EV-Monitor many of the open points mentioned above can then be clarified.

2. Concept

The overall concept of the EV-Monitor and GPS-Tracker is shown in Figure 1. The central part, EV-Monitor box is installed in the vehicle to be monitored.



Figure 1: Overview of the concept showing the central part (EV-Monitor box built into automobile), connection to external networks and graphic interface.

The EV-Monitor records the charging processes and the travel paths (including the GPS position and associated time stamp). The measured data is transmitted to a central evaluation unit.

One could then carry out longer term tests with a fleet of electrical vehicles and gather enough data to evaluate the additional load on the utility power grid.

Furthermore the speed, height and the state of the battery charge are also logged to allow inferences about driving behavior and the associated current consumption and battery status.

The development of the EV Monitor's hardware, firmware and software including the graphical evaluation software was performed within this project. The device is equipped with the central microcontroller AT90CAN128 as well as the GPS- and GSM-module. The microcontroller-firmware is a modular C-Code that can be updated for configuration purposes via serial interface. The PC-software is implemented in JAVA and is split in two major functions: connection to the telecommunication's server to retrieve the transmitted data, and post-processing/visualization of the measured data.

3. Implementation

The EV-Monitor has three main components: hardware (EV-Monitor box), microcontroller firmware (MCU C-Code) and PC software (JAVA Code). The functionality and realization of these components are described below.

The block diagram of the EV-Monitor box (hardware part) is shown in Figure 2.

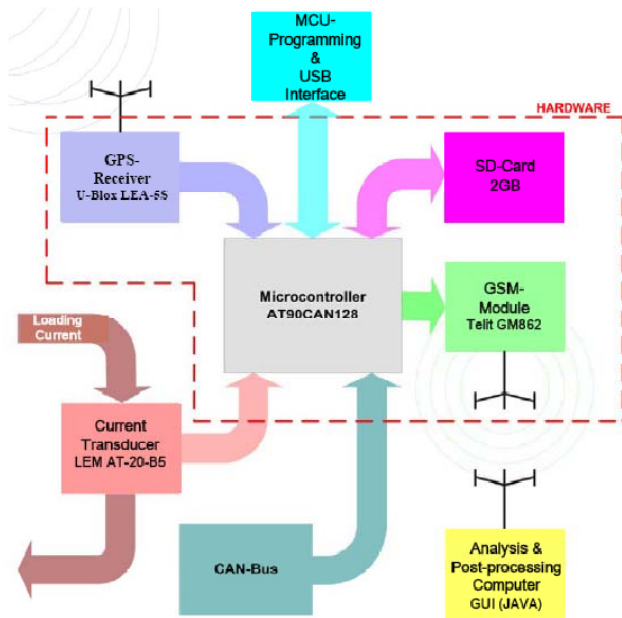


Figure 2: Block diagram of the hardware showing the kernel microcontroller and peripherals (GPS and GSM/GPRS modules and CAN, USB and sensor analog interfaces).

In the center is the Atmel microcontroller AT90CAN128 [1], chosen as a compromise of a small size/complexity MCU (16MIPS) still offering an integrated CAN-controller, three serial-interfaces (two UART and one SPI)

and built-in ADC (10 bits resolution) with multiplexed input for up to 8 channels.

The major electrical characteristics and adaptors for the hardware part are summarized in Table 1.

Table 1: Major electrical characteristics and adaptors.

Parameter	Symbol/ Adaptor	Min	Typ	Max
Supply/Consumption				
Current	I_{tot}	110 mA	125 mA	140 mA
Voltage	V_{tot}	5.0 V	12.0 V	14.8 V
Current Transducer Cinch female				
Voltage range		0.0 V		5.0 V
Input resistance		792 k Ω	800 k Ω	808 k Ω
GPS-Antenna SMA female				
Frequency range		1574 MHz	1575 MHz	1576 MHz
Gain			25 dB	27 dB
Impedance			50 Ω	
GSM-Antenna SMA female				
Frequency range		820 MHz	900 MHz	980 MHz
Gain				3 dBi
Impedance			50 Ω	
CAN-Interface	DSUB9 male			
USB	mini A			

The GPS-Receiver LEA-5S [2] from u-Blox is connected to one of the serial interfaces (UART) and deliver the selected GPS-data in the standard NMEA protocol. The reception is enabled via an external active antenna, which can be mounted in the front shield of the car.

The current transducer AT-20-B5 [3] from LEM requires no external supply and delivers a linear DC voltage output proportional to the measured RMS-value of the AC current input. The sensor output is connected to an input ADC channel of the MCU. The sensor type chosen for the 1st prototype has a maximum AC input current of 20A, but different types can be selected in case of carrying out measurements in loading points with higher capacities.

The CAN-Interface and integrated controller enable the reception of CAN-data made available by other car parts connected to this bus. In our case the biggest interest was to collect battery data as charge status, battery current, recuperation status, and the car speed. Many other data are available, but since the CAN-Bus protocol is not fixed, it is required to obtain from the manufacturer of each car part the required ID words to retrieve the information in the CAN-Bus traffic.

The GSM Module GM 862 [4] from Telit is used to establish a bidirectional wireless link with a telecommunication's server connected to internet. So far this connection has been used to send the collected GPS- and CAN-data for remote analysis and post-processing in a PC acting as the central evaluation unit. Future extensions will include remote services as for example consulting via SMS the battery status, the average consumption, and, receiving via SMS information like the location of load points located nearby.

The collected data is also locally stored in a SD-Card such that no data loss occurs in case the GSM connection is lost (for instance while charging the battery in an underground garage). The SD-Card has a capacity to store up to one year of measurement data (if sampling interval is kept to 8 s).

The initialization of the GPS and GSM modules, the decoding of the GPS- and CAN-data, the control of the measurement (triggering data acquisition), and the control of the data logging and data transmission (as presented in the previous paragraphs) are all handled by the microcontroller C-Code (MCU firmware).

Since the application is not time critical, most operations can be kept in the main loop and interrupts are mainly used for updating flags and variables, which trigger events in the main function. The code is split in dedicated functions for specific interfaces and tasks, and care has been taken to make it easily maintainable (e.g. numerical values kept in #define pre-compiler statements).

The logged data to be transmitted via GSM is packed into a telegram of 54 Bytes which contains besides the GPS- and the CAN-data, an identifier for the vehicle status (drive or load) and a vehicle ID-number.

A set of commands (analog to AT-commands) has been defined to enable configuration of certain firmware parameters (like ID-number, interval to send data via GSM) and to enable the read out of logged data in the SD-card. These commands and return data are sent and received via a PC-hyperterminal connected to the USB interface.

In case that the user wants to stop the data logging and transmission a private switch has been implemented for this purpose.

The PC-Software is split into a:

- JAVA RX-Server which connects via internet to the telecommunications-Server, retrieves the transmitted data and stores it into separated data files for each ID-number (<id_number>.edat).
- JAVA graphical user interface (GUI) based on the programming environment NetBeans, using extensions from swinglabs and chart resources from jFreeChart. The GUI is kept simple and can be operated in few mouse clicks (vide Figure 3).

The GUI reads the edat-files and displays the vehicle's trajectory and detailed information to each loading point. Out of the received data further diagrams relating consumption, speed and height can also be derived.

The GUI also enables to export the data from the edat-files into comma separated files (csv-files) for further analysis with other tools like Matlab and Excel (with better statistical resources).

4. Measurement Results

During a test phase of two weeks with an electrical vehicle of type "Twingo Elektra" the functionality of data acquisition, transmission and analysis of the EV-Monitor was successfully verified.

Every 8 seconds the EV-Monitor recorded the charging current, the battery charge, the vehicle speed, and the GPS-data including position, altitude, date and time. The measured data was successfully automatically sent to the servers at the Zurich University of Applied Sciences (ZHAW). There the developed graphical analyses tool display detailed charging curves as well as the driven route including altitude profile and battery charge state.

To give an example of the tests executed the graphical output of three measurement sequences are presented next.

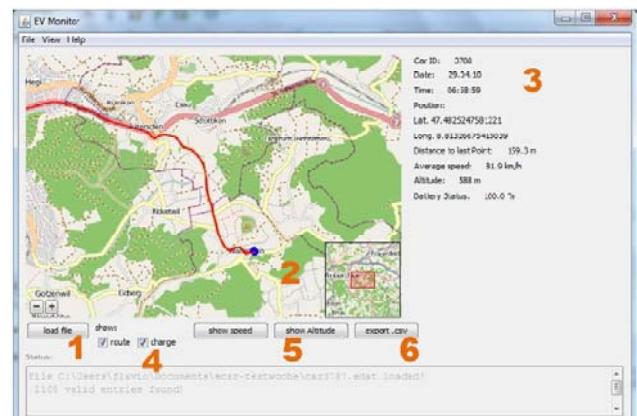


Figure 3: Graphical user interface for visualization of measured and transmitted data.

Figure 3 shows the GUI start window. First, indicated by number 1, the user should choose the edat-file to be imported. As soon as a file is imported the registered trajectory appears as a red line and the battery loading points as blue points.

Clicking at a loading point (number 2) causes the GUI to open another window showing the load curve of the battery (figure 4).

Pointing to the trajectory with the mouse will trigger the route information (number 3) to appear on the right upper side of the main GUI window. The switches indicated by number 4 enable to select the data to be displayed (route, charge or both).

The button pointed by number 5 triggers the opening of additional windows showing further diagrams with average and instantaneous speed versus time (figure 6) or height versus time (figure 7). These diagrams are combined with the battery status data for comparison and correlation.



Figure 4: Measurement of a complete loading cycle with instantaneous and average (over last 50 values) current plus associated battery status (charging from 12% to 91% over 16 hours).

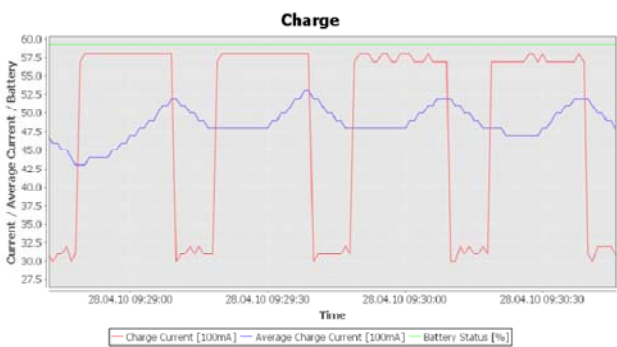


Figure 5: Detail of loading curve to observe current pulsing behavior during load phase.

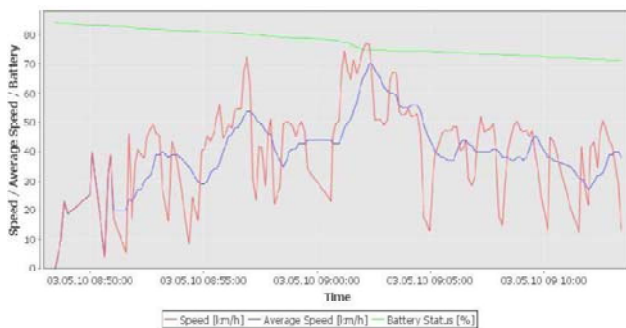


Figure 6: Instantaneous and average (over last 50 values) speed values and associated battery status for the 3rd test trajectory (without recuperation).

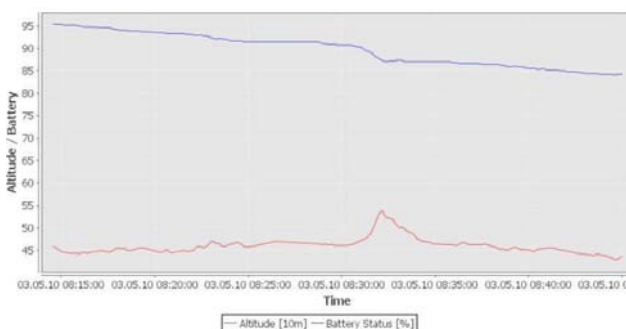


Figure 7: Height values and associated battery status for the 3rd test trajectory with activation of the battery recuperation.

The charging process can be observed in figures 4 and 5. The scale on the Y-axis represents the current values in 100mA and the battery charge status in %. In the control board of the EV the maximum load current was set to 7A.

Figure 4 show a complete charging cycle over 16 hours from almost empty to full battery status. Different phases can be identified, a pulsing load phase and later short reactivation for battery temperature control. The battery charge rises almost linearly except for a jump in the end, which by inspection of the log file seems to be due to lack of CAN-data for this time period.

Figure 5 shows a detail of the pulsing load phase of another test sequence (here the acquisition interval was reduced to 1 s). In this detailed view it can be observed that the loading current for this battery type tends to pulse with a width of about 2/3 between minimum and maximum values. Except for a short spike the loading current is kept below the maximum value set on the control board.

Figures 6 and 7 show further diagram possibilities for a 3rd test trajectory. Figure 6 shows the instantaneous and average speed in Km/h and for comparison the corresponding battery status. Figure 7 shows the height values and associated battery status.

This 3rd test trajectory was repeated with similar speed but having the battery recuperation on and off to evaluate the influence of the recuperation mechanism into the battery load. For this trajectory the recuperation system could reduce the battery load consumption by 11%.

The bill of material of the 1st prototype of the EV-Monitor box adds up to about 630 US dollars. Future evolution of the prototype (using a combined GPS and GSM module) and ordering parts in larger numbers can contribute to significant material cost reductions.

5. Conclusions and Outlook

The expansion of the EV market and the planning of the appropriate charging infrastructure require better knowledge than currently available about the users' habits and detailed battery behavior (loading and discharge profiles).

In order to measure and acquire these data an EV-Monitor and GPS-Tracker device was designed, implemented and tested in the Zurich University of Applied Sciences.

The analysis of the measured data can be carried out in any PC having internet access to the selected telecommunication's server. As a part of the project a graphical user interface was developed, which enables the visualization of the vehicle's trajectory, the loading spots, and for each loading spot the associated loading curve.

Field tests have been carried out and the results achieved demonstrate that the ZHAW EV-Monitor is ready to test a larger number of electric vehicles.

The loading curve of single EVs can be very interesting for battery and car producers seeking for information about performance of their product in real application scenarios.

And tests with a fleet of autos can be very informative for energy suppliers seeking to preview and balance the electricity demand and eventually integrate vehicle's batteries in smart grid systems.

Currently a 2nd prototype series is being produced for tests with the local energy supplier of the greater Zurich area (EKZ).

Future evolution of the EV-Monitor will include:

- hardware optimizations like usage of a combined GPS/GSM module, addition of new sensors (for example a luminosity sensor could be added to evaluate the potential of a photovoltaic solar roof) and optionally a display;
- firmware/software expansions to allow remote services with SMS (request battery status, indicate nearest charging station, etc);
- GUI extensions to show average consumption in KW/km, and enable comparison of two or more data-files.

Ideally the collected data will be embedded in a qualitative survey of the test persons' expectations before and their impressions after the test period, as well as an evaluation of the actual driving and charging behavior during the test period. This associated market study is also planned for a future project at the ZHAW.

6. Acknowledgements

We would like to thank the help and support of EKZ [5] (Urs Wiederkehr) who lend us the electrical vehicle used for the field tests, and of Kamoo [6] (Marc-Andre Beck) producer of this electrical vehicle, who supported us with technical issues on the CAN-bus.

7. References

- [1] [Atmel AT90CAN128 Datasheet Hyperlink](#)
- [2] [u-Blox LEA-5S Datasheet Hyperlink](#)
- [3] [LEM AT-20-B5 Datasheet Hyperlink](#)
- [4] [Telit GM862 Datasheet Hyperlink](#)
- [3] www.ekz.ch
- [4] www.kamoo.ch

8. Authors



Dr. Ing. Marina de Queiroz Tavares
Zürich University of Applied Sciences,
School of Engineering,
Professor of Signal Processing and
Telecommunications in the Department of
Electrical Engineering.
dqtm@zhaw.ch
<https://home.zhaw.ch/~dqtm/>



B.Tech Jonas Gartmann
Zürich University of Applied Sciences,
School of Engineering,
Graduating bachelor student in the
Department of Electrical Engineering.
j.o.n.a.s@gmx.ch



B.Tech Flavio Perucchi
Zürich University of Applied Sciences,
School of Engineering,
Graduating bachelor student in the
Department of Electrical Engineering.
flaonline@bluewin.ch



Prof. Dr. Franz Baumgartner
Zürich University of Applied Sciences,
School of Engineering,
Professor of Renewable Energy in the
Department of Electrical Engineering.
bauf@zhaw.ch
<https://home.zhaw.ch/~bauf/>



MSc. Maria Youssefzadeh
Zürich University of Applied Sciences,
School of Engineering,
Head of Center for Aviation and
Transport Systems.
youm@zhaw.ch