Demo: Wireless Battery Management System

Fabian A. Rincon Vija[1,2], Samuel Cregut[1], Georgios Z. Papadopoulos[2], Nicolas Montavont[2]

[1] Renault Group, Guyancourt, France

{fabian-antonio.rincon-vija, samuel.cregut}@renault.com

[2] IMT Atlantique, IRISA, France

{georgios.papadopoulos, nicolas.montavont}@imt-atlantique.fr

***Abstract*—Electric Vehicles (EV) are considered as the present and future of transportation. One of the most critical parts of EVs is the Battery Management System (BMS). Lately, the industry has seen the interest in replacing the traditional wired bus of the BMS with a wireless network as it brings several benefits. Considering that it is a critical application, the wireless network must meet strict requirements such as low energy consumption, bounded latency, and high reliability. In our previous works, we have proposed a Wireless BMS (WBMS) network based on the Group Acknowledgement (GACK) method and the Time Slotted Channel Hopping (TSCH) Medium Access Control (MAC) mode running over the Bluetooth Low Energy (BLE) physical layer to meet these requirements. The goal of this demonstration is to present a real-world implementation of our WBMS proposal integrated in a Renault Zoe to show the feasibility of this new application of low-power wireless networks.**

***Index Terms*—Electric Vehicles, EV, Battery Management System, Wireless BMS, IoT, Industrial IoT, TSCH.**

I. Introduction

EV have been an active subject of research and development over the years. Thanks to their several advantages, they are considered the present and future of transportation. The battery is one of the most critical parts of an EV. Typically, an EV battery pack is composed of cells grouped into modules. For example, the Renault Zoe battery pack is divided into 12 modules, each containing eight cells. To ensure a safe operation, the EV needs a BMS to monitor the battery cells' voltage and temperature. Keeping these values within specific intervals is crucial to avoiding fatal accidents and extending the battery lifetime. A BMS is composed of a master in charge of executing the necessary calculations and one slave per module, which measures the cells' voltage and temperature, see Fig. 1. When the car is in driving or charging state, the BMS is in active mode. During this active mode, the BMS slaves periodically measure and transmit the cell data to the master. The master computes the State of Charge (SOC) and the State of Health (SOH) of the battery pack using the data collected by the slaves. It also generates balancing orders to keep all the cells with the same charge level. When the EV is in parking state, the BMS should pass to sleep mode, where it only needs to keep the battery pack balanced.

In a traditional BMS, the master communicates with the slaves through a wired daisy chain Universal Asynchronous Receiver-Transmitter (UART). Lately, the industry has seen the interest in replacing this wired bus with a wireless link between the master and the slaves, as you can observe in Fig. 1. WBMS brings several benefits like simpler battery second life management,

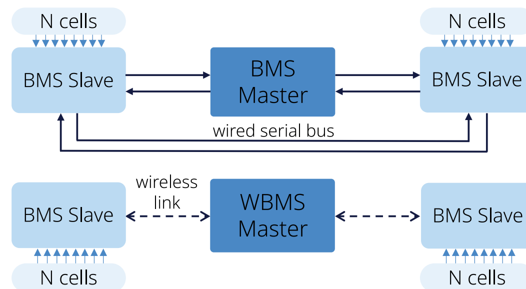

Fig. 1: From BMS to WBMS.

easier fabrication, and fewer connection failure risks. As the BMS is a critical system, the WBMS network must provide high reliability, bounded latency, and low energy consumption. The goal of this demonstration is to present a real implementation of a wireless BMS developed to perform driving tests in a Renault Zoe to show the feasibility of this new application of low-power wireless networks. The following sections detail our WBMS proposal and the developed test environment to validate it.

II. The proposed Wireless BMS

Replacing the wired bus on the BMS brings several advantages but, at the same time, multiple challenges. A WBMS network must ensure high reliability, bounded latency, and low energy consumption. In the WBMS network of a Renault Zoe, composed of 1 master and 12 slaves, the master needs to receive a new measure from all the cell voltage every 100 ms. In such a crucial system, the data Packets Delivery Ratio (PDR) should be higher than 99.999%, and the average current consumption of each node in active mode should be less than 1 mA. As it is proposed in [1], to meet the requirements, a TSCH-based network over the Bluetooth Low Energy (BLE) physical layer with a star network topology can be employed. The TSCH MAC mode, which is defined by the IEEE Std. 802.15.4 [2], uses time division to avoid message collision and frequency diversity to gain robustness against frequency interference [3]. This MAC protocol splits the time into small intervals called timeslots, which have sufficient length to allow a message transmission. Then it groups these timeslots into a bigger structure called slotframe, which repeats periodically over the time [4]. The message transmission for two consecutive timeslots occurs on different radio channels. How the network coordinates the message exchange between the nodes is defined

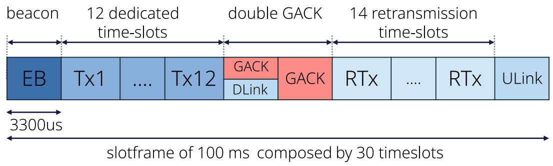

Fig. 2: TSCH schedule based on the GACK mechanism to achieve high reliability in the Renault Zoe WBMS network [1].

by a schedule, which indicates to the nodes what action they have to perform at each timeslot (transmit, receive or sleep) [4].

To achieve high reliability in the WBMS network, the master can orchestrate the data exchange with the nodes using a Group Acknowledgement (GACK) based schedule [1]. The idea of this schedule is to create a slotframe with the same length of the update data period, i.e., 100 ms for the Renault Zoe case. This slotframe is divided into two parts, the first for static transmissions and the second for dynamic retransmissions, as depicted in Fig. 2. Each node has a dedicated timeslot to send the battery cell voltage and the module temperature measure to the master in the first part. Then, the master broadcast two GACK messages reporting which frames were not received. Based on the GACK, the nodes know which timeslot they can use for the second part of the slotframe to retransmit their message if necessary. The main advantage of this method is that if a message is lost, the transmitter has more than one retransmission opportunities to successfully transmit the message. Note that the first timeslot of the slotframe is reserved for an Enhanced Beacon (EB) frame. The EB is used to synchronize the network and allow nodes to rejoin if they ever lose connection, so, for the Renault Zoe WBMS, the network synchronizes the nodes every 100 ms [1].

In sleep mode, if the master turns off, the slaves would lose synchronization, so they must continuously listen to the radio channel waiting for an eventual wake-up command from the master. This behavior would consume much energy and cannot guarantee a rapid wake-up time. On the other hand, keeping the master on to synchronize the network could discharge the small 12 V battery of the EV. Our WBMS proposal includes a sleep mode where one different node periodically broadcasts an EB to reduce the clock drift between the slaves. This method allows the master to be switched off until the vehicle goes to active mode again and to adapt the frequency at which slaves wake up, reducing their power consumption [5].

III. Demonstration

To validate our proposal, we have implemented a WBMS to run driving tests in a Renault Zoe. We use custom-made slaves, which contain a radio microcontroller (TI Simplelink cc2642) and an ASIC BMS (TI BQ79616). The first is a low-power MCU that manages the wireless communication with the master and the UART interface with the BMS ASIC. The TI BQ79616 measures up to 16 cell-voltage values with a precision of $\pm 1\, mV$; in the Renault Zoe case, each module has only eight cells. For the master, we use a TI Simplelink 26x2 Launchpad to manage the network, which has access to a CAN bus using a BOOSTXL-CANFD-LIN evaluation board. The CAN interface is necessary to send the battery data received from the nodes to a modified version of the Renault Zoe BMS, which allows the vehicle to move using our WBMS solution.

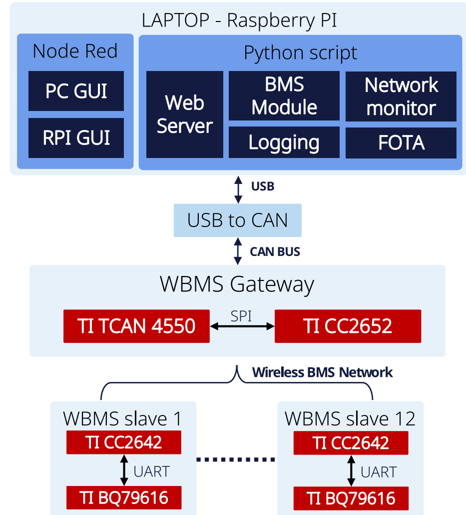

Fig. 3: WBMS demo setup.

Before integrating our system into the vehicle, we have done static tests. The results show that our proposal presents high reliability (100% with more than 60 million messages), deterministic latency (a new measurement received every 100 ms), and low-power consumption ($400\, \mu A$ in active mode and $40\, \mu A$ in sleep mode for the MCU in each slave). As presented in Fig. 3, our test setup includes a python script running on a laptop connected to the CAN bus through a USB-to-CAN device. This script receives the WBMS slaves measures from the master and the network data. Based on this information, the laptop shows the real-time voltages measures and the network statistics (PDR, channel stability, RSSI per node) using a GUI implemented in Node-red. The system allows us to test the network stability in sleep mode and even run over-the-air (OTA) updates in the network by sending a new firmware version over the CAN bus to the WBMS master.

Acknowledgment

We would like to thank Texas Instrument for providing all the necessary hardware used in this work.

References

[1] F. Rincon Vija, S. Cregut, G. Z. Papadopoulos, and N. Montavont, "Enabling Robust Wireless Communication for BMS on Electric Vehicles," in *IEEE 46th LCN*, 2021, pp. 423–426.

[2] "IEEE Standard for Low-Rate Wireless Networks," *IEEE Std 802.15.4, 2020*, pp. 1–800.

[3] T. Watteyne, A. Mehta, and K. Pister, "Reliability Through Frequency Diversity: Why Channel Hopping Makes Sense," in *6th ACM PE-WASUN symposium*, 2009.

[4] X. Vilajosana, T. Watteyne, T. Chang, M. Vučinić, S. Duquennoy, and P. Thubert, "IETF 6TiSCH: A Tutorial," *IEEE Communications Surveys & Tutorials 2019*, vol. 22, no. 1, pp. 595–615.

[5] F. Rincon Vija, S. Cregut, G. Z. Papadopoulos, and N. Montavont, "Autonomous Balancing Sleep Mode for Wireless BMS in Electric Vehicles," in *IEEE CSCN*, 2021, pp. 65–71.

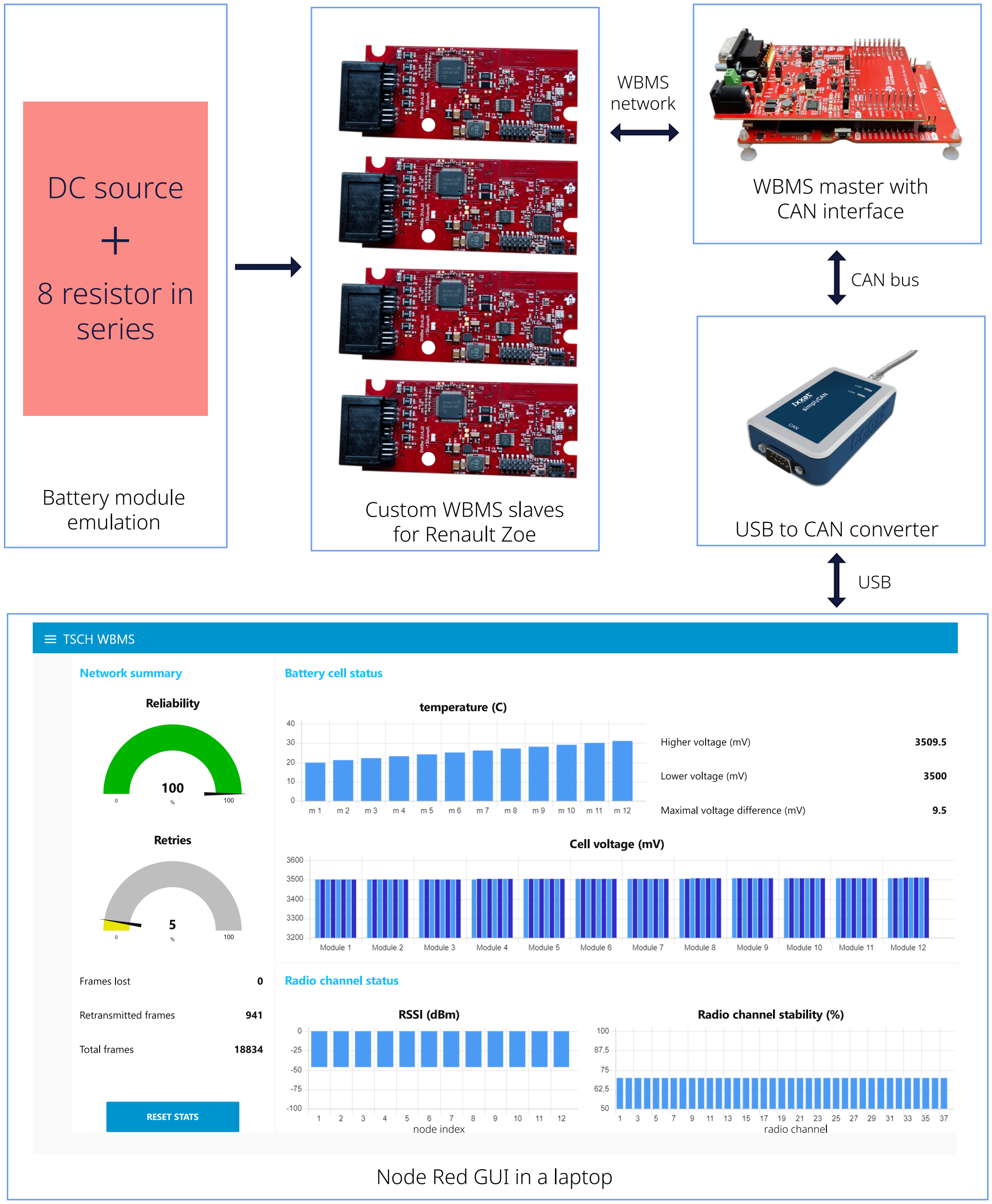

Fig. 4: WBMS demo proposal.

APPENDIX

A. Demonstration setup

For the demo, we will present a WBMS network with four nodes and one master. The WBMS slaves are connected to a custom board that uses eight resistors in series to emulate the battery cell voltages. A DC source powers this board. The nodes measure the emulated cell voltages and send them every 100 ms to the master through our proposed wireless WBMS network. The master receives the network data and sends it to the python script using the CAN bus. Once the software has received the CAN messages, a screen will display the real-time network statistics (PDR, channel stability, RSSI per node) and the emulated battery cell voltages, as shown in Fig. 4. The demonstration will also exhibit the OTA characteristics of our WBMS network and the sleep mode, which allow us to switch off the master completely without interrupting the network synchronization.

B. Requirements

This demo requires the following equipment:

- A table to put a laptop, a Raspberry PI, one external screen, and an area equivalent to two A3 sheets of paper for the WBMS devices with the battery module emulator.
- One external monitor, which could be standing on the table, or on a foot with a HDMI cable.