Smart Home Power Management System for EV Battery Charger and Electrical Appliance Control

Ruben E. Figueiredo[1,*], Vitor Monteiro[1], Joao C. Ferreira[3,4], Joao L. Afonso[1] and Jose A. Afonso[2]

[1] ALGORITMI Research Centre, University of Minho, 4800-058 Guimarães, Portugal

[2] CMEMS-UMinho, University of Minho, 4800-058 Guimarães, Portugal

[3] ISTAR-IUL, Instituto Universitário de Lisboa (ISCTE-IUL), Lisbon, 1649-026, Portugal

[4] Inov Inesc Inovação – Instituto de Novas Tecnologias, Portugal

Correspondence

Ruben E. Figueiredo, University of Minho, Campus de Azurem, Portugal

Email: ruben.figueiredo@algoritmi.uminho.pt

Summary: This paper presents a power management system (PMS) designed for smart homes aiming to deal with the new challenges imposed by the proliferation of plug-in electric vehicles (EVs) and their coexistence with other residential electrical appliances. The PMS is based on a hybrid wireless network architecture composed by a local hub/gateway and several Bluetooth Low Energy (BLE) and Wi-Fi sensor/actuator devices. These wireless devices are used to transfer information inside the smart home using the MQTT (Message Queuing Telemetry Transport) protocol. Based on the proposed solution, the current consumption of the EV battery charger and other residential electrical appliances are dynamically monitored and controlled by using a configurable algorithm, ensuring that the total current consumption does not cause the tripping of the home circuit breaker. An Android client application allows the user to monitor and configure the system operation in real-time, a developed Wi Fi smart plug permits to measure the RMS values of current of the connected electrical appliance and change its state of operation remotely, and an EV battery charger may be controlled in terms of operating power according to set-points received from the Android client application. Experimental tests are used to evaluate the quality of service provided by the developed smart home platform in terms of communication delay and reliability. An experimental validation for different conditions of operation of the proposed smart home PMS concerning the power operation of the EV battery charger with the proposed control algorithm is also presented.

KEYWORKS: smart home, power management system, electric vehicle, wireless network, MQTT.

List of Symbols and Abbreviations:

Δt	desired charging time (h)
ADC	Analog-to-Digital Converter
AP	Access Point
BLE	Bluetooth Low Energy
CR	available battery capacity (kWh)
CT	total battery capacity (kWh)
DBMS	Database Management System
DSM	Demand-Side Management
EV	Electric Vehicle
EVBC	Electric Vehicle Battery Charger
FC	Fixed Current (charging method)
GUI	Graphical User Interface
I2C	Inter-Integrated Circuit
I_{EA}	RMS value of the electrical appliances current (A)
I_{EVBC}	RMS value of the EVBC current (A)
I_H	RMS value of the total current consumed at home (A)
I_{MAX}	RMS nominal value of the home circuit breaker current (A)
IoT	Internet of Things
IP	Internet Protocol
ISM	Industrial, Scientific and Medical
ISP	Internet Service Provider
MAC	Medium Access Control
MQTT	Message Queuing Telemetry Transport
PHY	physical layer
PM	Power Management
PMS	Power Management System
QoS	Quality of Service
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver-Transmitter
USB	Universal Serial Bus
VC	Variable Current (charging method)
V_c	battery charging voltage (V)
WPAN	Wireless Personal Area Network

1. INTRODUCTION

The electric vehicle (EV) is recognized as a strategic element for the new paradigm of mobility in the transport sector.[1,2] Moreover, by adopting dedicated control strategies for the EV charging process, it can also provide a relevant contribution to the integration of renewable energy sources.[3] Consequently, it also contributes for the sustainable development of smart grids,[4,5] which are facing emerging challenges and opportunities due to the EV introduction.[6] This is even more relevant in the context of the integration between smart homes and smart grids. As an example, a three-port integrated topology used for interfacing an EV and a set of solar photovoltaic panels with the electrical power grid, at residential power level, is proposed and experimentally validated in [7]. In this context, aiming an effective proliferation of EVs, new technological aspects are emerging regarding bidirectional controlled operation (namely the concepts of grid-to-vehicle, G2V, and vehicle-to-grid, V2G),[8] as well as bidirectional communication with EVs.[9–11] Based on the strategic introduction of EVs with bidirectional power operation (G2V and V2G) and bidirectional communication, new perspectives can be offered for EV users and for the power management (PM) in the perspective of smart homes and smart grids.[12] Moreover, besides the bidirectional power operation, other possibilities of operation can be addressed by the EVs as a contribution for improving the reliable operation of smart homes and smart grids, as demonstrated in [13] and [14].

Taking into account that the EV will be plugged-in at residential level for performing the battery charging process, smart homes also play an essential role in the development of smart grids.[15] In this context, a control plan for the EV operation based on the power grid energy price variation and the EV uncertainties is proposed in [16], and the V2G operation for the power grid in terms of ancillary services, considering the energy price and the EV battery state-of-charge, is introduced in [17] and [18]. Consequently, the provision of new operation modes will be fundamental for the EV operation in smart homes and their integration with smart grids, as demonstrated in [19] and [20]. In this sense, the implementation of the Internet of Things (IoT) paradigm in the context of smart homes is essential to address the new challenges posed by the EV proliferation.[21]

Figure 1 presents an application scenario of a power management system (PMS) in a smart home based on a wireless network infrastructure. As can be seen, this system allows the monitoring and control of both the EV battery charger (EVBC) and the residential electrical appliances, enabling the provision of innovative PM features. In this context, the smart home platform, which provides wireless communication between the EVBC and the residential electrical appliances, is absolutely essential for providing the innovative PM features. This paper is framed in this scope, where a smart home PMS is proposed, allowing to control the operation of the EVBC and other residential electrical appliances, and taking into account the user preferences through an Android client application. More specifically, the proposed smart home PMS is responsible for monitoring the total current consumed at home (I_H), and for sending this information, in real-time, to the gateway, where the PM algorithm is executed. As consequence, the PM algorithm sends on/off commands to specific residential electrical appliances, as well as dynamic commands to the EVBC in order to maximizing the charging current (I_{EVBC}) while preventing the tripping of the main circuit breaker. Besides the proposed application scenario, the developed smart home platform may be simultaneously used to provide other services, such as monitoring and control of the heating, lighting, and security systems.

[Figure 1]

Summarizing, the main contributions of this paper are: (i) The possibility to configure dynamic set-points of bidirectional power operation, regarding the implemented solution for the EVBC, compared with a traditional EVBC with bidirectional functionalities that does not allow the dynamic operation; (ii) A solution for monitoring and dynamic control the electrical appliances of the smart home, which is based on non-invasive methods through the development of a Wi-Fi smart plug; (iii) An innovative control algorithm that ensures that the RMS value of the EVBC current plus the RMS value of the electrical appliances current, do not exceed the nominal value of the home circuit breaker current, avoiding the actuation of protection systems as occurs with traditional solutions; (iv) Experimental validation for different conditions of operation of the proposed smart home PMS based on developed laboratory prototypes, including the validation of the communication and the power operation of the EVBC with the proposed control algorithm.

Regarding the main design approaches that were taken into account in the development of the smart home platform, it can be highlighted: (i) The implementation of a hybrid Wi-Fi/BLE solution designed for the smart home PMS, due to the versatility to support different wireless technologies; (ii) The use of open-source software components and tools; (iii) The use of standard communication protocols; (iv) The versatility to adapt to different applications and associated sensor/actuator devices (e.g., taking into account distinct power consumptions and data rate requirements); (v) The reliable operation, even in case of failure of the connection to the Internet; (vi) The flexibility to integrate other network technologies; (vii) The minimization of the number of required hardware components (and, consequently, of the cost) through the integration, in the same device, of several functionalities provided by the platform; (viii) The very low cost of the wireless modules incorporated in the EVBC and in the residential electrical appliances, in comparison with the equipment where they are incorporated.

This paper is organized as follows. Section 2 presents a summary and discussion of related work. The developed smart home PMS, including the PM algorithm, the architecture, functionalities and hardware costs of its smart home platform, is described in section 3. The main experimental results for validating both the performance of the smart home platform and the operation of the PMS are provided in section 4. Finally, the conclusions are presented in section 5.

2. RELATED WORK

This section aims to situate the proposed smart home PM system with respect to similar related work, highlighting the key advantages and drawbacks of each system and clarifying the main contributions of this paper.

In [22], the author presents a smart home system architecture and a prototype with distributed and embedded flexible edge analytics for home energy management and demand-side management (DSM). Similarity to this paper, our proposed smart home PMS uses the concept of edge computing, as the control algorithm runs locally and not in the cloud, due to the reasons stated in the paper, namely: (i) Internet network connectivity may not always be available or be limited; (ii) this approach reduces network latency, contributing to better real-time responsiveness. On the other hand, the system presented in the paper uses a Raspberry Pi as Wi-Fi access point, whereas we use a separated access point, normally provided by the Internet Service Provider (ISP), which allows handling the IoT traffic and the conventional Internet traffic with the same equipment. In our system, we use a Raspberry Pi to execute the proposed PM algorithm, as well as performing several other features, such as: BLE/Wi-Fi gateway, MQTT broker, HTTP server and local database. The NodeMCU Wi-Fi module is used in both prototypes, but our smart home architecture also integrates BLE modules, which, unlike Wi-Fi modules, are able to provide low energy consumption, being suitable for energy-constrained sensor devices.

In [23], the authors propose an PMS conceived to incorporate the residential sector into DSM and to facilitate the integration between renewable energy sources (RES) and energy storage systems (ESS). The proposed algorithms dynamically perform the scheduling of household appliances and ESS, based on the pricing of electricity market, in order to minimize the electricity bill and reduce the peak-to-average ratio (PAR). The paper compares the performance of different types of optimization algorithms proposed to solve this problem through MATLAB simulations, and also reviews other DSM strategies proposed for minimization of electricity cost, reduction of PAR, improvement of power system efficiency and/or mitigation of carbon emissions. The smart home PMS proposed in this paper may be used to provide the communication and computing infrastructure required for this and several other DSM algorithms proposed in the literature, such as [24].

In [25], the authors propose a smart home platform that uses three different wireless network technologies inside the house: ZigBee, Bluetooth (classic) and IEEE 802.11b. The rationale, like in our platform, is that each technology may complement the others in other to overcome their disadvantages. The transmitter used in this test was a Raspberry Pi connected to an XBee 900HP ZigBee module, whereas the receiver was a STM32L1 microcontroller attached to another XBee 900HP module. Unlike this platform, the BLE and Wi-Fi modules used in our platform include the microcontroller and transceiver hardware in the same printed circuit board (PCB), contributing to decrease its size and cost.

In [26], the authors present results demonstrating that BLE, which is integrated in the proposed smart home PMS, is more energy efficient in terms of number of bytes transferred per energy spent than IEEE 802.15.4/ZigBee, concluding that BLE consumes less energy. Another advantage of BLE over ZigBee is that it offers higher data rate at the physical layer (1 Mbps vs. 250 kbps). BLE is also a better alternative compared to classic Bluetooth, which has higher energy consumption and is limited to a maximum of eight devices per network, unlike BLE.[27]

3. POWER MANAGEMENT SYSTEM

This section starts with the presentation of the proposed algorithm for the smart home PMS, which is followed by the description of the smart home platform that was developed to provide support for the PMS communication and computing tasks. Finally, it describes the developed smart plug prototype, which enables the monitoring and control of the current consumption of the residential electrical appliances. It is important to highlight that the main objective of the power management strategy is related with the possibility to configure dynamic set-points of operation for the EVBC, as well as the monitoring and dynamic control of electrical appliances of the smart home (based on non-invasive methods through the development of a Wi-Fi smart plug), supported by an innovative control algorithm. As main result, it is guaranteed that the RMS value of the EVBC current plus the RMS value of the electrical appliances current, do not exceed the nominal value of the home circuit breaker current, avoiding the actuation of protection systems as occurs with traditional solutions.

3.1 PM Algorithm

The architecture of the proposed smart home PMS defines two main types of electrical equipment that are controllable using the developed smart home platform: (i) An EVBC whose operating power is controlled according to real-time measurements of the home power demand; (ii) A set of controllable residential electrical appliances. This section describes the methodology of operation of the smart home PMS, with emphasis on the control of the EVBC.

As single-phase electrical installations are the most common for smart home scenarios, the proposed control is for this type of installations, aiming to prevent the main circuit breaker tripping. The control algorithm ensures that the RMS value of the EVBC current (I_{EVBC}), plus the RMS value of the electrical appliances current (I_{EA}), do not exceed the nominal value of the home circuit breaker current (I_{MAX}). In this sense, the relation between I_{EVBC}, I_{EA}, I_{MAX} and the total current consumed in the home (I_H) is expressed by the following equations:

$$I_H = I_{EVBC} + I_{EA} \quad (1)$$

$$I_H \leq I_{MAX} \quad (2)$$

Analyzing (1) and (2), it is possible to recognize that the EVBC current (I_{EVBC}) and the current consumed by the electrical appliances (I_{EA}), must be updated in real-time, otherwise, the total home current (I_H) can exceed the maximum allowed current (I_{MAX}). Concerning the EVBC current (I_{EVBC}), two charging methods were defined: VC (Variable Current) and FC (Fixed Current). The VC method is adopted when the user does not define a specific charging time. In this case, the EVBC current is adjusted according to the available current (i.e., as a function of the current consumed by the electrical appliances). On the other hand, the FC method is adopted when the user requires to charge the EV battery within a certain time. In this case, the charging current is fixed, being determined at the beginning of the charging process by:

$$I_{EVBC} = \frac{(C_T - C_R)}{V_c \, \Delta t} \times 1000 \qquad (3)$$

where C_T is the total battery capacity (kWh), C_R the available battery capacity (kWh), V_c the charging voltage (V), and Δt the desired charging time (hours). In this circumstance, the EVBC current is imposed, while the current consumed by the set of electrical appliances must be adjusted according to the remaining available current.

The control algorithm, represented in Figure 2, is based on determining whether the total current of the home is above or below the maximum allowed value (I_{MAX}). If I_H is lower than I_{MAX}, the state of the EVBC is checked. If it is on and the charging method is VC, the charging current is increased in order to take advantage of all the available current, to enable faster charging. On the other hand, if I_H is higher than I_{MAX}, the device that caused this increase is checked by looking at the last one that was turned on (the EVBC or a residential appliance), in order to determined how to deal with the overload. If the VC charging method is active, its current is reduced. Otherwise, if the overload has been caused by the EVBC and the FC charging method is active, the control algorithm turns off appliances previously defined with lower priority, one by one, until there is enough current, whereas if the device that caused the overload was a residential appliance, this device is turned off. In all cases, whenever a device is turned off by the algorithm, the user is notified.

[Figure 2]

3.2 Smart Home Platform

The implementation of protocols from the Internet protocol suite[28] is required for a sensor device to be able to connect directly to the IoT. This includes the Internet Protocol (IP), at the network layer, as well as a transport layer protocol, such as Transmission Control Protocol (TCP) or User Datagram Protocol (UDP), and an application layer protocol, such as HTTP. These protocols are implemented in Wi-Fi devices, along with specific lower protocols for the physical (PHY) layer and MAC (Medium Access Control) layer defined on the IEEE 802.11 standard.[29]

On the other hand, wireless personal area network (WPAN) technologies such as BLE and IEEE 802.15.4/ZigBee, being optimized for low power operation, implement their own protocols, which are not directly compatible with the Internet protocol suite (although there are proposals to enable direct connection of IEEE 802.15.4 networks to the Internet through the use of the 6LoWPAN protocol, which compresses the IPv6 header to make it suitable for low power wireless networks.[30] Therefore, the developed smart home platform includes a local hub/gateway device that is responsible to translate the exchanged packets between the specific WPAN protocols and the Internet protocols, allowing the integration of these WPAN-based sensor devices into the IoT. The integration of a gateway in the platform also allows the provision of other benefits. For example, the implementation of a security firewall in the gateway to monitor and control the communications may protect the sensor devices from malicious attacks.[31] In this sense, the gateway is an important part of the architecture of the proposed smart home platform, which is represented in Figure 3.

[Figure 3]

3.2.1 Local Gateway

The gateway, identified in the middle left of the figure, was implemented using a Raspberry Pi 4 Model B, which is a small and low-cost single-board computer that provides dual-band wireless local area network (2.4 GHz and 5 GHz) compatible with the IEEE 802.11.b/g/n/ac standards, Bluetooth 5.0 (which supports BLE), Ethernet, USB 2.0 and USB 3.0 ports, among other features. We used its main operating system (OS), called Raspbian, for the development of this platform prototype. Raspbian is a free OS based on Debian Linux and optimized for the Raspberry Pi.

The developed platform was designed for autonomous and continuous operation even in case of failure of connection to the Internet. In this sense, the gateway also acts as the central brain of the system, running processes of interest for the operation of the smart home, such as automation rules/algorithms, and providing a local database where relevant information is stored and can be accessed using the client application, developed for an Android smartphone (presented in section 3.2.4).

The message exchange between the devices of the proposed smart home PMS is handled by the MQTT protocol. MQTT is an application layer protocol that is suitable for resource-constrained devices that may use low data rate or unreliable links.[32] MQTT uses the publish/subscribe model to provide implementation flexibility and simplicity, handling the message exchange even if the IP addresses of the devices in the network change along the time. The main components of an MQTT network are the publishers, the subscribers and a broker. The publishers are entities of the network that generate data, such as sensor devices. The subscribers, on the other hand, are entities interested in receiving data from particular topics, such as actuator devices. The broker manages the registrations for specific topics from publishers and subscribers. After that, the publisher may transmit information to the interested subscribers through the broker.[32] The MQTT broker was implemented in the Raspberry Pi through the installation of Eclipse Mosquitto,[33] which is an open source and lightweight message broker, with the MQTT 3.1.1 protocol. The MQTT clients (publishers and subscribers) can be installed in the Wi-Fi devices, but not in the BLE devices, because, as referred before, these devices do not use the Internet protocol suite. In order to avoid the introduction of another device in the platform to handle this problem, it was developed a program performing functions of a gateway BLE/Wi-Fi. The MQTT client required for publishing the data generated by the BLE devices was implemented in the Raspberry Pi, through the installation of a Python module called paho-mqtt[34] and the Python module used to implement the BLE client to handle the wireless communication with the BLE devices was bluepy.[35]

MQTT defines three levels of quality of service (QoS),[36] with different degrees of reliability: at most once (0), at least once (1), and exactly once (2). In QoS level 0, the MQTT client sends each data message only once, and there is no checking if the message was delivered, so there is the possibility that the message is lost. QoS level 2 uses an acknowledgement message called PUBACK to confirm the delivery of the data message. When the client does not receive the acknowledgement, it retransmits the message. With this QoS level 1, it is possible to receive a duplicate copy of the data message if the PUBACK message is lost. QoS level 2 uses a 4-way handshake mechanism to ensure that the data message is delivered exactly once, providing the higher reliability level, but possibly with higher end-to-end delays compared to the other levels, due to the exchange of four messages.

For data storage, a local database was implemented in the Raspberry Pi using MariaDB,[37] an open source database management system (DBMS) based on MySQL. This database stores all relevant data that passes through the MQTT broker. A static IP address was assigned to the Raspberry Pi to ensure that its address is not changed over time, allowing access to the database (through the HTTP server) and the MQTT broker from a known address.

3.2.2 Wireless Router

The wireless router performs two main functions: acting as the house's Wi-Fi access point (AP) and allowing the connection of this local network to the Internet. The wireless router equipment is specific to the user's house and is normally supplied by his/her Internet Service Provider (ISP). The IoT traffic shares the house's Wi-Fi network with the conventional Internet traffic, avoiding the requirement of a supplementary network infrastructure. The wireless router used in this paper is a Huawei HG8247Q, supplied by the Vodafone Portugal ISP. This router supports port forwarding, which allows outside devices to connect to devices inside the local network. With this mechanism, the client application may access the services provided by the local gateway, namely the database and the MQTT broker, from the Internet.

Through the connection to the Internet, the developed platform may also use the IoT services provided by an external cloud platform, such as ThingSpeak,[38] Google Cloud Platform,[39] IBM Bluemix,[40] Amazon Web Services (AWS) or Microsoft Azure,[41] for storing and processing IoT data.

3.2.3 Sensor/Actuator Devices

Each sensor and/or actuator device in the developed platform is mainly composed of two parts: A wireless module (Wi-Fi or BLE) and a sensor/actuator component or equipment. The wireless modules are general-purpose boards that may be used with different sensors and/or actuators and applications. They integrate a microcontroller for data processing and I/O (Input/Output) interfacing with the sensors/actuators, a wireless network transceiver, an antenna and other auxiliary components in the same PCB. Some sensors provide an analog signal that is acquired and processed by the microcontroller using an analog-to-digital converter (ADC), while others already provide digital output signals using a standard serial interface protocol, such as UART (Universal Asynchronous Receiver-Transmitter), SPI (Serial Peripheral Interface) or I2C (Inter-Integrated Circuit). The wireless module may also be connected to a more complex equipment, such as the EVBC used in this paper. In this sense, the proposed smart home platform was conceived to be generic and versatile in order to allow its adaptation to a vast range of applications.

Although Figure 3 only represents two sensor/actuator devices, the proposed smart home PMS allows the integration of several Wi-Fi devices (directly connected to the wireless router) and BLE devices (indirectly connected through the gateway). In fact, in the developed prototype, the EVBC and the smart plugs (for the residential electrical appliances) are connected to different Wi-Fi devices.

The Wi-Fi module used in this prototype is the NodeMCU development board, due to its low cost, open source firmware and ease of programming through the Arduino integrated development environment (IDE). This module provides ADCs, supports multiple digital data interfaces (I2C, SPI and UART) and it is capable to implement an MQTT client.

The BLE module used in this platform was the FireBeetle ESP32 development board, due to its low cost, low power consumption and robust design. It supports the ADC, SPI, I2C and UART interfaces and includes a Wi-Fi transceiver, if needed. A BLE server was implemented through the Arduino IDE to send the data collected from the attached sensors.

Although the developed platform prototype is based on BLE and Wi-Fi networks, it has flexibility to allow the introduction of other wired/wireless network technologies. For example, a ZigBee network can be integrated into the platform using a ZigBee coordinator board (such as the one developed in [42]) attached to the Raspberry Pi through one the available USB (Universal Serial Bus) ports, allowing the reception of sensor data from ZigBee devices. The data interfaces provided by the Raspberry Pi allow also the communication with other wireless or wired devices using technologies such as LoRa,[43] IEEE 802.3/Ethernet or PLC (Power Line Communication).[44] Furthermore, since all the wireless network technologies referred above operate in unlicensed ISM (Industrial, Scientific and Medical) frequency bands there are no costs associated with the utilization of the network to transfer data, unlike the choice of using cellular network technologies.

The proposed smart home PM system must be able to provide real-time communication services for the envisioned applications when required. For example, the total elapsed time from the data acquisition from a sensor device until the delivery of the information to an actuator device (eventually passing through intermediate devices responsible for routing, processing/control algorithms and/or storage), must be under the desired delay bound. In order to validate this feature, experimental results concerning the delay associated with the operation of the developed smart home platform in the context of the power management system are presented in section 4.1.

3.2.4 Client Application

The developed client application provides a graphical user interface (GUI) to the smart home power management system, allowing the user to monitor and control its operation. The mobile phone sensing paradigm[45,46] also allows a smartphone to become a sensor device, providing readings from its integrated sensors (e.g., accelerometer or microphone), or to act as another gateway, allowing Bluetooth/BLE sensor devices (e.g., wearable heart rate monitors) to send information to the system. The client application was implemented for an Android smartphone and integrates an MQTT client, so that it can receive and send data. In Figure 4 is represented the main Android activity, which works as the main application menu and allows access to the different house areas (Garage, Ground Floor, First Floor) and to the home energy management section.

[Figure 4]

3.3 Smart Plug Prototype

In order to allow the monitoring and control of the residential electrical appliances, which is required by the proposed smart home PMS, a smart plug was developed. Each developed smart plug allows controlling the state (on/off) of the electrical appliance connected to it, as well as reporting its current and voltage values through the implementation of an MQTT client on a Wi-Fi module.

A NodeMCU board (section 3.2.3) was used for data acquisition, local processing and Wi-Fi communication. The analog current sensor DFRobot 20 A was used to measure the AC current, whereas the analog voltage sensor ZMPT101B was used to measure the AC voltage. A 5 V relay module was used to control the state (on/off) of the connected appliance. Since the NodeMCU has only one ADC input, it was necessary to add an analog multiplexer to the project. The resulting circuit diagram is represented in Figure 5, and the smart plug prototype with identification of its main components is presented in Figure 6.

[Figure 5]

[Figure 6]

3.4 Cost Analysis

Table 1 lists the individual cost of each component of the proposed system, as well as its overall cost. This analysis considered the cost of a single Wi-Fi Smart plug (55.05 €) and a single sensor for measuring the home current, but the total cost will depend on the number of smart plugs used at the smart home.

[Table 1]

The costs presented in Table 1 are based on retail prices for low quantities from a single seller and include sales taxes, but lower costs could be achieved with wholesale purchases and the selection of sellers with lower prices. Some components could also be replaced by other similar but cheaper products. For example, the Firebeetle ESP32 could be replaced by a similar ESP32-DevKitC-32D board, which costs 13.60 € from the same seller.

4. RESULTS AND DISCUSSION

This section presents experimental results obtained from tests performed for the overall system, comprising the performance evaluation of the smart home platform, the validation of the proposed PMS, as well as results concerning the developed Android client application.

4.1 Performance Evaluation of the Smart Home Platform

The experimental performance tests presented in this section are based on the hardware described in section 3.2 and the platform architecture represented in Figure 3. The tests evaluate two relevant QoS parameters in the context of operation of the PM algorithm: Delay and reliability (expressed by the delivery ratio). In this sense, the tests measure the values of these parameters in the communication path between a sensor device and the respective actuator device. As an example, the sensor device (MQTT publisher) may be the home current sensor, whereas the actuator device (MQTT subscriber) may be the EVBC or a controllable electrical appliance. We performed these tests for two different types of sensor devices: a BLE module (peripheral device), represented by a FireBeetle ESP32 board; and a Wi-Fi module, represented by a NodeMCU board. The potential advantage of using a BLE module is the lower power consumption, which is particularly relevant with battery-operated sensor nodes. However, in the considered application scenario, the sensor device is not energy constrained, because it has easy access to the mains power near the attached home current sensor; therefore, the use of a Wi-Fi module is a viable alternative. In all tests, the actuator device is a NodeMCU board. The tests were performed using the Mosquitto broker version 1.6.3, which was the latest version available at the time of the tests. For the NodeMCU MTTQ client, we used the arduino-mqtt library version 2.4.3,[47] whereas for the Raspberry Pi MQTT client we used the Eclipse Paho Python Client.[34] The tests were executed using MQTT QoS level 1 and 2, since they provide high degree of reliability.

For each of the following tests, 1000 samples were generated at the sensor device and sent to the actuator device. The measured delay corresponds to the time elapsed since the data is sent by the source endpoint (start time) until it is received in the destination endpoint (end time). Since the endpoints are located in different devices, they do not share the same clock. Therefore, in order to enable the measurement of the delay, each time, the source device sends a data packet, it sends an interrupt signal to the destination device, allowing this device to calculate the difference between the end time and the start time using the same clock. The source device waits the interrupt signal (confirmation that the previous packet was delivered) to send the next data packet. In order to evaluate the communication reliability, a sequence number was included inside each data packet sent by the source and verified in the destination. In the following tests, all application-level data packets arrived at the destination, which means that the delivery ratio in all tests was 100 %.

Regarding the delay in the path between the sensor device and the actuator device with QoS 1 and QoS 2, Figure 7 represents the delay results in the form of cumulative distribution functions, whereas Table 2 and Table 3 present the delay statistics (arithmetic mean, minimum and maximum). Two different configurations were tested. In Table 2, the sensor device is a Wi-Fi module and the path of information is from the MQTT client (publisher) in the sensor device to the MQTT broker in the gateway (Raspberry Pi), and from that to the MQTT client (subscriber) in the actuator device. In Table 3, the sensor device is a BLE module and the path of information starts with a link from the BLE server in the BLE module (BLE peripheral device) to the BLE client in the Raspberry Pi (BLE central device). Inside the same application, the data is transferred from the BLE client to the MQTT client (publisher). Then the data is sent to the MQTT broker in the Raspberry Pi, and from that to the MQTT subscriber in the actuator device, via Wi-Fi. In both configurations, the data packets have to pass through the wireless router, which means that they have to cross four Wi-Fi links.

[Figure 7]

[Table 2]

[Table 3]

Through the analysis of the results obtained, it is possible to verify that the minimum and average delay with the QoS 1 level (4.011 ms and 9.783 ms) are roughly half of those obtained with the QoS 2 level (7.612 ms and 16.548 ms) in the configuration where only Wi- Fi is used. These results are explained by the fact that QoS 1 requires the exchange of half of the packets in comparison with QoS 2. On the other hand, for the configuration where the sensor device is a BLE module, the average delay with the QoS 1 (52.567 ms) is only slightly lower than the value with QoS 2 (53.501 ms). Moreover, the configuration based only on Wi-Fi presents lower average delays (9.783 ms and 16.548 ms) than the hybrid BLE/Wi-Fi network configuration (52.567 ms and 53.501 ms). These results can be explained by two facts: (i) The communication path for the hybrid configuration is longer; (ii) On average, the access delay and the transmission delay for BLE packets is higher than for Wi-Fi packets. The consequence is that the delay in the BLE link, when present, contributes to most of the total delay.

The maximum delay values observed, in all cases, were less than 500 ms. These values are sufficiently low for the smart home PMS to react to changes in the home current value, preventing the tripping of circuit breaker and allowing the user to interact with the system without significant lags. Regarding the MQTT level to be used in the system, it is advisable to use QoS 2, because although the QoS1 delay times are shorter, duplicate messages can occur, which means that the final receiver should implement additional logic to detect duplicates that might affect the behavior of the system.

4.2 Validation of the PMS

The results obtained during the experimental validation regarding the control of both the EVBC and the electrical appliances in the smart home are presented in this section. These results comprise distinct validations. The tests were carried-out in laboratory considering realist conditions and with a power grid voltage value of 230 V - 50 Hz and with an operating power value of 3.45 kW. Figure 8 shows the laboratory workbench for testing the controllability of the EVBC and the electrical appliances.

[Figure 8]

Figure 9 shows a key experimental result concerning the RMS values of the home current (I_H), the EVBC current (I_{EVBC}) and the electrical appliances current (I_{EA}) when the VC charging method is selected. As shown, during case #1 the EVBC is still not operating, but some electrical appliances are turned on, increasing the current consumption of the home; for this reason, the home current (I_H) is equal to the current consumed by the electrical appliances (I_{EA}). Nevertheless, I_H is still lower than I_{MAX}, so the proposed algorithm does not take action. In the second case (#2), the EVBC starts its operation and the current consumption of the electrical appliances remains the same. Therefore, due to the proposed strategy for adjusting the EVBC consumption, the current of the EVBC (I_{EVBC}) is defined according to the equations (1) and (2), i.e., it is the difference between the maximum allowed by the circuit breaker (I_{MAX}) and the current of the electrical appliances (I_{EA}). At the beginning of case #2, I_{EA} has a value close to I_{MAX}; consequently, the EVBC is allowed to have only a small current consumption (I_{EVBC}) according to the proposed algorithm. At the beginning of case #3, an electrical appliance was turned off by itself; therefore, the current I_{EA} is reduced with a sudden variation, allowing I_{EVBC} to increase through the intervention of the proposed algorithm, but still keeping the total current below I_{MAX} in order to avoid the tripping of the circuit breaker, i.e., in this case, the sudden increase of I_{EVBC} is equal to the sudden decrease of I_{EA}, where the current $\mathrm{I_{EVBC}}$ is the controlled variable. In the last case (#4), all the electrical appliances are turned off (an operation that corresponds to situations where all the electrical appliances finish performing their tasks by themselves), resulting in increasing the current of the EVBC (I_{EVBC}) until the maximum allowed value (I_{MAX}), which is limited by the circuit breaker.

[Figure 9]

Figure 10 shows a key experimental result concerning the RMS values of the home current (I_H), the EVBC current (I_{EVBC}) and the electrical appliances current (I_{EA}) when the FC charging method is selected. The first case (#1) is similar to the first case of Figure 9 and, therefore, the home current (I_H) is equal to the current consumed by some electrical appliances (I_{EA}). In the second case (#2), the EVBC starts its operation with a fixed value of current obtained through equation (3), which leads to a I_H superior than the I_{MAX} allowed. The PM algorithm detecting this overload sends commands to turn off, one by one, the electrical appliances with lower priority than the EVBC, until I_H is equal or less than I_{MAX}.

[Figure 10]

4.3 Android Client Application

The results obtained during the experimental validation of the developed Android application are presented in this section. Figure 11 demonstrates the operation based on the FC method, which corresponds to the case where the user requires EV battery charging within a specified time. In (a) the desired time was set to 8 hours, and in (b) the related data are presented: battery capacity; charging voltage; charging current; time remaining; expected completion; and a graph of I_{EVBC} over time. In this case, the value of I_{EVBC}, which was calculated through the equation (3), is constant (9.24 A), since the FC method is based on the use of a fixed charging current.

[Figure 11]

Figure 12 demonstrates the operation of the smart home PMS using the VC method, which means that the EVBC current is adjusted according to the available current at home. In (a), it is shown the energy management screen, where it is visible the I_{EA} value, the available current at home before the EVBC was turned on, and the I_{EVBC} value, as well as the current consumption percentage by home division. Since I_{MAX} was set to 15 A and I_{EA} was 11.9 A, the initial current assigned to I_{EVBC} was approximately 3 A, as shown in the line graph at (b). During this test, variations in I_{EA} were made, causing the graph to display different values for I_{EVBC} over time.

[Figure 12]

5. CONCLUSION

The increasing replacement of internal combustion engine vehicles by electric vehicles imposes challenges in the EV battery charging process, particularly at home, since the parking periods when the charging occurs are concurrent with the operation of other electrical appliances. In this context, this paper presents a smart home power management system (PMS), based on the use of IoT technologies, for monitoring and controlling the EV battery charging process and the operation of other electrical appliances. The main contributions of this paper are the proposal of a PMS supported by the developed smart home platform and the proposed control algorithm, which enable the distributed control the EVBC and other residential electrical appliances. The design goals in the development of the smart home platform included the use of low-cost hardware, the use of open-source software and tools, the selection of standard communication protocols, and the flexibility to integrate other network technologies.

All components of the developed smart home PMS were validated through the execution of experimental tests. The measured delay values were suitable in the context of the application considered in this paper, allowing the power management system to adjust the current available for the EVBC, avoiding the home circuit breaker tripping. The PM algorithm behaved as expected in all planned situations, the developed smartphone application allowed easy monitoring and control of the EVBC and electrical appliances, and the developed smart plug was fundamental to obtain the RMS values of current and to change the state of each electrical appliance.

CONFLICT OF INTEREST STATEMENT

The authors declare no conflict of interest.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

1. Raghavan SS, Khaligh A. Electrification potential factor: Energy-based value proposition analysis of plug-in hybrid electric vehicles. *IEEE Trans Veh Technol*. 2012;61(3):1052-1059. doi:10.1109/TVT.2011.2181438
2. Salmasi FR. Control strategies for hybrid electric vehicles: Evolution, classification, comparison, and future trends. *IEEE Trans Veh Technol*. 2007;56(5 I):2393-2404. doi:10.1109/TVT.2007.899933
3. Humfrey H, Sun H, Jiang J. Dynamic charging of electric vehicles integrating

renewable energy: a multi-objective optimisation problem. *IET Smart Grid*. 2019;2(2):250-259. doi:10.1049/iet-stg.2018.0066

4. Gao S, Chau KT, Liu C, Wu D, Chan CC. Integrated energy management of plug-in electric vehicles in power grid with renewables. *IEEE Trans Veh Technol*. 2014;63(7):3019-3027. doi:10.1109/TVT.2014.2316153
5. Saber AY, Venayagamoorthy GK. Plug-in vehicles and renewable energy sources for cost and emission reductions. *IEEE Trans Ind Electron*. 2011;58(4):1229-1238. doi:10.1109/TIE.2010.2047828
6. Monteiro V, Afonso J, Ferreira J, Afonso J. Vehicle Electrification: New Challenges and Opportunities for Smart Grids. *Energies*. 2018;12(1):118. doi:10.3390/en12010118
7. Monteiro V, Pinto JG, Afonso JL. Experimental Validation of a Three-Port Integrated Topology to Interface Electric Vehicles and Renewables with the Electrical Grid. *IEEE Trans Ind Informatics*. 2018;14(6):2364-2374. doi:10.1109/TII.2018.2818174
8. Monteiro V, Ferreira JC, Nogueiras Melendez AA, Couto C, Afonso JL. Experimental Validation of a Novel Architecture Based on a Dual-Stage Converter for Off-Board Fast Battery Chargers of Electric Vehicles. *IEEE Trans Veh Technol*. 2018;67(2):1000-1011. doi:10.1109/TVT.2017.2755545
9. Khaligh A, Li Z. Battery, ultracapacitor, fuel cell, and hybrid energy storage systems for electric, hybrid electric, fuel cell, and plug-in hybrid electric vehicles: State of the art. *IEEE Trans Veh Technol*. 2010;59(6):2806-2814. doi:10.1109/TVT.2010.2047877
10. Liu N, Chen J, Zhu L, Zhang J, He Y. A key management scheme for secure communications of advanced metering infrastructure in smart grid. *IEEE Trans Ind Electron*. 2013;60(10):4746-4756. doi:10.1109/TIE.2012.2216237
11. Escudero-Garzas JJ, Garcia-Armada A, Seco-Granados G. Fair design of plug-in electric vehicles aggregator for V2G regulation. *IEEE Trans Veh Technol*. 2012;61(8):3406-3419. doi:10.1109/TVT.2012.2212218
12. Wirasingha SG, Emadi A. Classification and review of control strategies for plug-in hybrid electric vehicles. *IEEE Trans Veh Technol*. 2011;60(1):111-122. doi:10.1109/TVT.2010.2090178
13. Monteiro V, Pinto JG, Afonso JL. Operation Modes for the Electric Vehicle in Smart Grids and Smart Homes: Present and Proposed Modes. *IEEE Trans Veh Technol*. 2016;65(3):1007-1020. doi:10.1109/TVT.2015.2481005
14. Monteiro V, Pinto JG, Afonso JL. Improved vehicle-for-grid (iV4G) mode: Novel operation mode for EVs battery chargers in smart grids. *Int J Electr Power Energy Syst*. 2019;110:579-587. doi:10.1016/j.ijepes.2019.03.049
15. Gungor VC, Sahin D, Kocak T, et al. Smart grid and smart homes: Key players and pilot projects. *IEEE Ind Electron Mag*. 2012;6(4):18-34. doi:10.1109/MIE.2012.2207489
16. Zhang T, Chen W, Han Z, Cao Z. Charging scheduling of electric vehicles with local renewable energy under uncertain electric vehicle arrival and grid power price. *IEEE Trans Veh Technol*. 2014;63(6):2600-2612. doi:10.1109/TVT.2013.2295591
17. Smolenski R, Jarnut M, Benysek G, Kempski A. AC/DC/DC interfaces for V2G applications-EMC issues. *IEEE Trans Ind Electron*. 2013;60(3):930-935. doi:10.1109/TIE.2012.2188876
18. Galus MD, Koch S, Andersson G. Provision of load frequency control by PHEVs, controllable loads, and a cogeneration unit. *IEEE Trans Ind Electron*. 2011;58(10):4568-4582. doi:10.1109/TIE.2011.2107715
19. Liu C, Chau KT, Wu D, Gao S. Opportunities and challenges of vehicle-to-home, vehicle-to-vehicle, and vehicle-to-grid technologies. *Proc IEEE*. 2013;101(11):2409-2427. doi:10.1109/JPROC.2013.2271951

20. Jin C, Tang J, Ghosh P. Optimizing electric vehicle charging: A customer's perspective. *IEEE Trans Veh Technol*. 2013;62(7):2919-2927. doi:10.1109/TVT.2013.2251023
21. Al-Fuqaha A, Guizani M, Mohammadi M, Aledhari M, Ayyash M. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Commun Surv Tutorials*. 2015;17(4):2347-2376. doi:10.1109/COMST.2015.2444095
22. Lin YH. Novel smart home system architecture facilitated with distributed and embedded flexible edge analytics in demand-side management. *Int Trans Electr Energy Syst*. 2019;29(6):1-21. doi:10.1002/2050-7038.12014
23. Ahmad A, Khan A, Javaid N, et al. An Optimized Home Energy Management System with Integrated Renewable Energy and Storage Resources. *Energies*. 2017;10(4):549. doi:10.3390/en10040549
24. Kazmi S, Javaid N, Mughal MJ, Akbar M, Ahmed SH, Alrajeh N. Towards optimization of metaheuristic algorithms for IoT enabled smart homes targeting balanced demand and supply of energy. *IEEE Access*. 2017;7:24267-24281. doi:10.1109/ACCESS.2017.2763624
25. Adiono T, Putra RVW, Fathany MY, Wibisono MA, Adijarto W. Smart home platform based on optimized wireless sensor network protocol and scalable architecture. In: *Proceeding of the 2015 9th International Conference on Telecommunication Systems Services and Applications, TSSA 2015*. Institute of Electrical and Electronics Engineers Inc.; 2016. doi:10.1109/TSSA.2015.7440441
26. Siekkinen M, Hiienkari M, Nurminen JK, Nieminen J. How low energy is bluetooth low energy? Comparative measurements with ZigBee/802.15.4. In: *2012 IEEE Wireless Communications and Networking Conference Workshops, WCNCW 2012*. ; 2012:232-237. doi:10.1109/WCNCW.2012.6215496
27. Bluetooth Special Interest Group. *Specification of the Bluetooth System. Master Table of Contents & Compliance Requirements, Version 5.0*.; 2016.
28. Tanenbaum, Andrew S; Wetherall DJ. *Computer Networks: Pearson New International Edition*.; 2014.
29. IEEE 802.11-2016 - 802.11-2016 - IEEE Standard for Information technology—Telecommunications and information exchange between systems Local and metropolitan area networks—Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications | Joinup. https://joinup.ec.europa.eu/solution/ieee-80211-2016-80211-2016-ieee-standard-information-technology-telecommunications-and-information. Accessed April 19, 2020.
30. Culler D, Chakrabarti S. 6LoWPAN: Incorporating IEEE 802.15. 4 into the IP architecture. *White Pap*. 2009.
31. Rowland C, Goodman E, Charlier M, Lui A, Light A. *Designing Connected Products: UX for the Consumer Internet of Things*. O'Reilly Media; 2015.
32. Al-Fuqaha A, Guizani M, Mohammadi M, Aledhari M, Ayyash M. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Commun Surv Tutorials*. 2015;17(4):2347-2376. doi:10.1109/COMST.2015.2444095
33. Eclipse Foundation. Eclipse Mosquitto. https://mosquitto.org/. Accessed January 4, 2019.
34. Eclipse Paho - MQTT and MQTT-SN software. https://www.eclipse.org/paho/clients/python/. Accessed April 23, 2020.
35. Harvey I. Python interface to Bluetooth LE on Linux. https://github.com/IanHarvey/bluepy. Accessed July 15, 2019.
36. Lee S, Kim H, Hong DK, Ju H. Correlation analysis of MQTT loss and delay according to QoS level. In: *International Conference on Information*

Networking. ; 2013:714-717. doi:10.1109/ICOIN.2013.6496715
37. MariaDB Foundation. About MariaDB - MariaDB.org. https://mariadb.org/about/. Accessed February 2, 2019.
38. Maureira MAG, Oldenhof D, Teernstra L. *ThingSpeak-an API and Web Service for the Internet of Things*. https://mediatechnology.leiden.edu/images/uploads/docs/wt2014_thingspeak.pdf. Accessed May 9, 2020.
39. Afonso JA, Sousa RA, Ferreira JC, Monteiro V, Pedrosa D, Afonso JL. IoT system for anytime/anywhere monitoring and control of vehicles' parameters. In: *Proceedings - 2017 IEEE International Conference on Service Operations and Logistics, and Informatics, SOLI 2017*. Vol 2017-January. Institute of Electrical and Electronics Engineers Inc.; 2017:193-198. doi:10.1109/SOLI.2017.8120993
40. Ferreira J, Afonso J, Monteiro V, Afonso J. An Energy Management Platform for Public Buildings. *Electronics*. 2018;7(11):294. doi:10.3390/electronics7110294
41. Ray PP. A survey of IoT cloud platforms. *Futur Comput Informatics J*. 2016;1(1-2):35-46. doi:10.1016/j.fcij.2017.02.001
42. Castro P, Afonso JL, Afonso JA. A low-cost ZigBee-based wireless industrial automation system. In: *Lecture Notes in Electrical Engineering*. Vol 402. Springer Verlag; 2017:739-749. doi:10.1007/978-3-319-43671-5_62
43. Raza U, Kulkarni P, Sooriyabandara M. Low Power Wide Area Networks: An Overview. *IEEE Commun Surv Tutorials*. 2017;19(2):855-873. doi:10.1109/COMST.2017.2652320
44. Majumder A, Caffery J. Power line communication: An overview. *IEEE Potentials*. 2004;23(4):4-8. doi:10.1109/MP.2004.1343222
45. Macias E, Suarez A, Lloret J. Mobile Sensing Systems. *Sensors*. 2013;13(12):17292-17321. doi:10.3390/s131217292
46. Khan WZ, Xiang Y, Aalsalem MY, Arshad Q. Mobile phone sensing systems: A survey. *IEEE Commun Surv Tutorials*. 2013;15(1):402-427. doi:10.1109/SURV.2012.031412.00077
47. Gähwiler J. MQTT library for Arduino. https://github.com/256dpi/arduino-mqtt. Accessed March 10, 2019.

Table 1. Cost of the proposed solution with the individual cost of each part.

Hub/ gateway	Raspberry Pi 4 (2GB RAM):	41.95 €
Wi-Fi Smart plug	NodeMCU ESP8266	7.50 €
	DFRobot 20 A	20.95 €
	ZMPT101B	13.15 €
	5 V Relay Module	2.45 €
	Analog Multiplexer 4051	1 €
	Box	5 €
	Male and Female Plug	5 €
Sensor device	Firebeetle ESP32	19.90 €
EVBC	NodeMCU ESP8266	7.50 €
	Total cost	**124.40 €**

Table 2. Delay statistics obtained for the Wi-Fi/Wi-Fi configuration: QoS 1 and 2.

Sensor Node: NodeMCU | Actuator Node: NodeMCU				
	Min (ms)	**Mean (ms)**	**Max (ms)**	**Packet loss (%)**
QoS 1	4.011	9.783	81.530	0%
QoS 2	7.612	16.548	307.626	0%

Table 3. Delay statistics obtained for the BLE/Wi-Fi configuration: QoS 1 and 2.

Sensor Node: ESP32 (using BLE) | Actuator Node: NodeMCU				
	Min (ms)	**Mean (ms)**	**Max (ms)**	**Packet loss (%)**
QoS 1	12.227	52.567	285.068	0%
QoS 2	23.335	53.501	453.786	0%