i-ETC
Telecommunications and Computers
Vol. 5 , n. 1 (2019) ID-1

ElectroGes - a Household IoT Energy Management System

Mariana Bento[a], Gonçalo Rodrigues[a], Vitor Vaz da Silva[ab],

[a]Department of Engineering of Electronics, Telecommunication and Computers,
ISEL/IPL - Instituto Superior de Engenharia de Lisboa, Portugal

[b]CTS-Uninova, FCT-UNL, Portugal

vsilva @ deetc.isel.ipl.pt

***Abstract[1]*— Domestic appliances plugged to the electrical mains power may, at some point, exceed the maximum contracted power supplied from the electrical company. To avoid a power cut-off on these cases, an energy management system is needed so that some appliances are automatically switched off before others are switched on. In order to do this, appliances have to be described in a privileged structure so that power is always available for some of them while other share leftovers. ElecroGes is a system that provides the house owner with a means to define the priorities of the connected appliances and manage them in an autonomous way; a working implementation is provided.**

Keywords: IoT, energy management, home, domotics

I. INTRODUCTION

The appliances found in a home are mainly electronic devices aimed at domestic usage and most of them are connected to the national, or private, energy supply network through an electrical plug. These devices present a consumption of energy over time defined as the power usage of the device while operational.

In Portugal, the electrical installations at each home, and the contracts with the electrical supply companies, limit the total domestic power consumption to a given value, which can lead to an energy cut, in the eventuality of all the connected devices consumption exceeds that limit [1].

In the Portuguese market, it is already possible to have access to several energy control systems such as the smart plug 'edimax SP-20101W' with which it is possible to manage the energy consumption of each device trough an iPhone or Android mobile device. Another example, specific for the Portuguese market, is the 'EDP re:dy plug' that transforms the connected appliances into smart ones and allows their management through a mobile device. As a final example, there is also a controller called 'Fator Potência PFW01-M12' which monitorizes eletrical measurements and can reduce or eliminate losses in an electrical system.

The ElectroGes system differs from the previous examples primarily because it joins the electrical consumption statistics with a configurable autonomous energy management system, without overloading the user. The objective of this project is to present a system which can reduce domestic power consumptions and avoid electrical cuts or failures due to excess, it also helps reducing the ecological impact and the financial cost of energy usage at home [2].

Even though ElectroGes presents a slow initial configuration process, once that is finished, the system does not require more user intervention, except to add new appliances or to change the desired settings.

II. ARCHITECTURE

The ElectroGes system is characterized by three main elements, these being the Controller, the EG-Plug (ElectroGes Plug) and the configuration App, all connected through the local Wi-Fi network [3][4] as shown in

Figure 1.

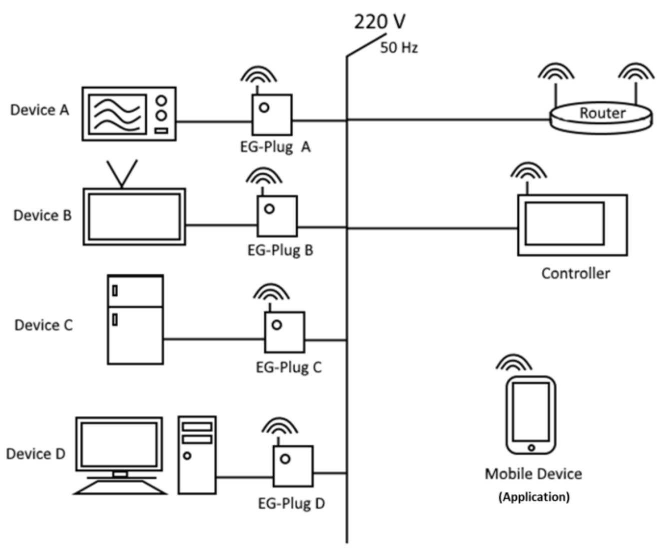

Figure 1 - Example diagram of the ElectroGes system

The Controller is the central processing and coordination element in the system. It reads the consumptions and manages all the connected EG-Plugs.

The EG-Plug is the link between the electrical network and the connected appliance.

Finally, the App is the tool used to make the initial configurations of each new EG-Plug introduced and allows its' connection to the system.

Both the Controller and the EG-Plug have an Hardware and a Software component as follows.

A. Hardware

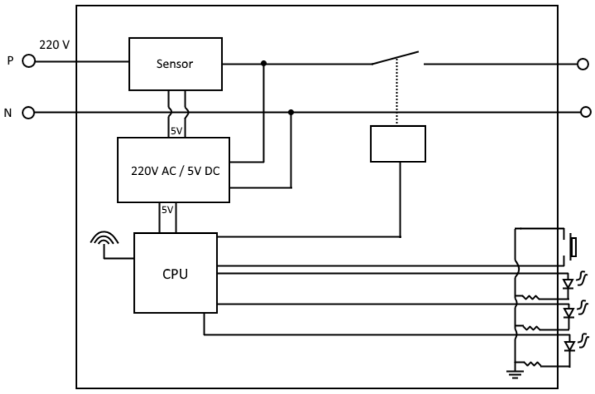

Figure 2 - Hardware diagram for the EG-Plug

The EG-Plug, shown in Figure 2, is essentially composed of a sensor block, which reads the amount of current being consumed using an Hall effect sensor (ACS712), a 220[V] AC to 5[V] DC transformer block, to power all the required components and a CPU block, where the entire processing is executed.

To implement the processor, the microcontroller NodeMCU ESP8266 was chosen due to the embedded integration of the Wi-Fi module ESP-12E [5].

To control the electrical current for the appliance, the Plug makes usage of an SSR (Solid-State Relay) but can also be used with an EMR (Electromechanical Relay).

The final aspect of the Plug's hardware is its user interface, with three LEDs, informing of the state of the Plug, and a push button that allows the user to manually override the module.

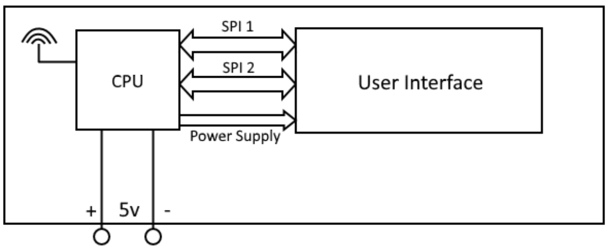

Figure 3 - Hardware diagram for the Controller

The Controller, shown in Figure 3, is simpler than the EG-Plug, presenting only a CPU block, which is equal to the CPU block used before, making use of the same microcontroller, and a user interface, implemented through the use of a small LCD screen with a resistive touch sensor and the respective driver.

In the prototype version of the Controller, the microcontroller is powered by the USB port, found on the ESP8266 board, and a regular smartphone charger.

The communication between the user interface and the microcontroller is made using two SPI (Serial Peripheral Interface) slave connections where the first channel communicates with the LCD screen and the second communicates with the screen's touch sensor driver, which converts the analog signals from the sensor, into SPI data frames.

B. Software

In structural terms, the Controller and EG-Plug have similar software, since it was developed in a hierarchic block structure. Each block is responsible for managing a corresponding hardware component, with a top layer main block which coordinates the remaining ones as a whole.

The main differences are visible in the blocks themselves as these differ from the EG-Plug to the Controller since they have distinct objectives and functionalities. For example, the block that has to control the SSR state only exists in the EG-Plug.

As mentioned above, the system communicates through the local Wi-Fi network, making use of the MQTT (Message Queueing Telemetry Transport) [6] and TCP/IP protocols to effectively transmit its data. The data is all formatted in a simple protocol, with specific fields for information and action parameters, created for the system and called EGC (ElectroGes Command).

III. APPLICATION & DISCUSSION

The boot up processes of both the Controller and the EG-Plug are very distinct. During the Plug's power up, a software AP (Access Point) is started creating a private network to allow for a device to connect and send the configuration data through the external app.

After the information reaches the Plug, it turns off the AP mode and attempts to connect to the specified Wi-Fi network from which it enters its normal operation mode, then being able to receive and send data to the connected Controller and communicate through the user interface.

Figure 4 - Frontal panel and user interface of the EG-Plug

As mentioned in a previous section, the user interface has three informative LEDs and a push button, identified as 'USER' on the Plug's panel as shown in Figure 4. Two more buttons exist in the prototype's interface, and not shown in Figure 2 to assist in debug and development; these are identified as 'RST' for resetting the hardware and 'FSH' to enable the program to be written to the flash memory.

The red LED informs the state of the plug, if it is on or off.

The blue LED in the middle informs the user if the Plug is connected to the Wi-Fi or not, depending whether it is on or off, and blinks if the Plug is receiving data.

Inside the system, each Plug has a priority level from 0 to 2 which is meant for situations where the system is required to shut down devices, and has to decide which ones to close. This way it can search for the lower priority (0) ones first, and only if no lower priority devices are active, the highest are going to be deactivated. When all devices are on the same priority level, the system turns off the ones which consume the most, considering that the level of priority 2, means that the device is never to be deactivated by the automatic system.

With this in mind, the final blue LED informs the user of the priority level of the Plug concerned, by the frequency at which the LED blinks. A low frequency indicates the low priority (0), a high frequency is used for high priority (1) and a constant light means the highest value of priority (2).

The button is used to manually override some of the Plug functionalities, which are separated by the pressing time on the button. A quick press toggles the electrical current on the Plug on or off, a short duration press, of 1 second, activates the LEDs for a 5 second period. All except for the Plug state LED are off to reduce idle energy consumption.

Finally a longer press, of 2 seconds, advances trough the priority levels in a looping way.

The Controller presents an interface that allows for the configurations to be edited in the device itself, not having the need to initiate an AP network on boot up, as shown on Figure 5.

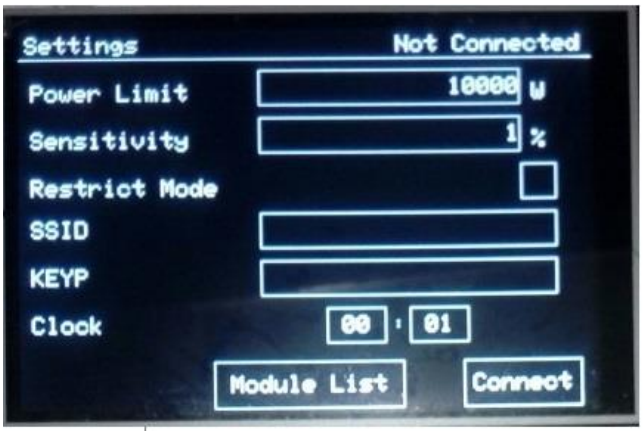

Figure 5 - Boot up and settings page of the Controller

Still on Figure 5 it is possible to see several fields such as 'Power Limit', which represents the total power allowed to be consumed by the sum of all Plugs connected to this Controller and the value in field 'Sensitivity', which represents the percentage of the maximum power that the real consumption value can oscillate around depending on if the 'Restrict Mode' is toggled.

When 'Restrict Mode' is on, the system starts disabling Plugs as soon as the total power they are consuming reaches the 'Max Power' minus the 'Sensitivity'. In case of it being off, then the system only intervenes when the total consumption rises above the 'Max Power' plus the 'Sensitivity'.

Finally, there is the 'Clock', which lets the user configure the current hour and minutes, allowing it to save the consumption statistics on the correct time they were registered. Since the system uses no external clock at this prototype stage, when the Controller resets, the clock resets to 00h00.

In order to connect the Controller to the network, its information has to be written in the settings page of the Controller on the 'SSID' (Network name) and the 'KEYP' (Network password) fields as shown in Figure 6. After being connected, the Controller will look for all the enabled Controllers in the network and assume itself the next available Id. When no Controllers are found, it assumes Id 0.

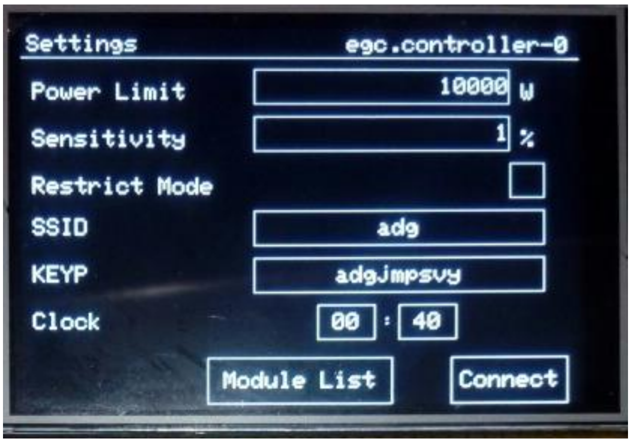

Figure 6 - Wi-Fi credentials on the setings page of the Controller

Both fields are saved in flash memory so that when the microcontroller resets it tries to reconnect to the network automatically. If the network is not available it is necessary to reintroduce the credentials of the desired network.

After the Controller is connected to the network and configured as intended, it is capable of accepting new EG-Plugs, connected through the use of the external configuration app and integrates them immediately into the system.

The initial configuration of each EG-Plug is shown in the Figure 7.

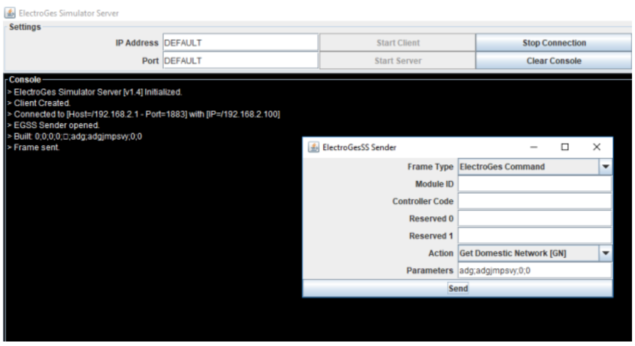

Figure 7 - Prototype app while configurating the first EG-Plug into Controller ID 0

In one of the Controller's pages the user can see the list of connected Plugs as shown in Figure 8. In this page it is possible to turn on or off any of the visible Plugs by clicking the green toggle button, which indicates the current state of each Plug (on/off). It is also possible to check the consumption statistics of each Plug by pressing the 'St' button.

In the test performed for the purpose of this article, the name of the connected Plug was left on default, which is "Module/" plus the internal ID for the Plug, being 0 the ID of the first Plug added for this Controller.

Figure 8 - Controller's Module (EG-Plug) list

In this page, it is also possible to configure each Plug and observe its immediate consumption by pressing the name of the module, opening the settings of the Plug concerned as shown in Figure 9 for the enabled module in which a 600[W] hairdryer with 2 levels of intensity, was connected on the first level, which consumed about 300[W].

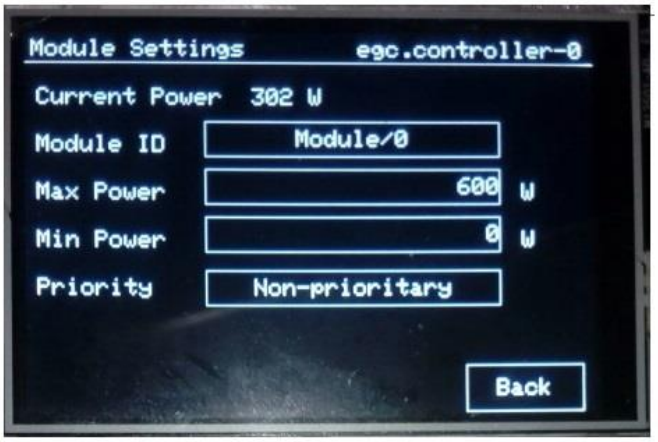

Figure 9 - Controller's Configuration menu for a specific EG-Plug

Other tests were made to ensure that the devices work as expected.

IV. FUTURE WORK

The final prototype still presents some imperfections such as some reading errors caused by interferences on the sensor and the methodology with which the data is obtained and read. These reading errors result in several small flaws like the power offset value of 30[W] when no current is flowing to the appliance and the value read should be of 0[W]. Another factor that influences the readings, when the appliance power is off, is the small amount of current that can flow through the SSR which has a value of roughly 25[mA] and causes an unwanted idle consumption of about 5[W] to 6[W].

A random delay in the order of milliseconds to two seconds has been noted while the system is working and a change of state is needed to switch off low priority devices; due to Wi-Fi access.

There is also a problem, related to the system integrity; when the Controller loses power, every connected Plug becomes "lose" and has to be reset as well and reconfigured to be reintegrated into the system, since in the prototype version there are no mechanisms to automatically restore the Plugs into the correct Controller. This can be solved by storing the configuration on the flash memory.

The mobile application is also necessary for user convenience, and it is also on the "to do" list.

V. CONCLUSIONS

Management of appliances at home, by switching off those that are considered less important that the one that is going to be switched on, is effectively done by this system, which allows the home inhabitants to have a more comfortable life with continuous access to their electrical energy supply.

Configuration by the users is set at a minimum so that an intuitive process is achieved.

The system can be scaled up by adding more EG-plugs while using the same Controller.

This project was developed in a considerably short period of time. In spite of that, given the evolution level the project achieved, during public presentation, it was considered, by the evaluating jury, as a prototyping-ready system. The success of the project lays on its functionality, as it was capable of correctly managing a single EG-Plug inside the given parameters without issues.

For this reason it was chosen to be presented in two different exhibitions, one of them meant for the future engineering students and the other showing 'The best of ISEL' to ISEL students.

There are some plans and ideas that would be interesting to develop and implement in order to improve the system further: First of all, to better organize the hardware components and reduce the EG-Plug size.

Next, to overcome the memory constrains imposed by the chosen microcontroller, since they limit the user interface design capabilities and the number of EG-Plugs that can be connected to a single Controller.

Still related to the Controller, the user interface should be updated in order to present a better design and to allow for more features to be added, like a lock screen or consumption graphics.

Further development of the external App, could connect it to a desired Controller first and obtain all the needed information to automatically configure a desired EG-Plug, thus facilitating the initial setup for the user.

Finally creating a scheduling system to allow the user to choose dates and hours in which specific EG-Plugs turn on or off., this we believe is a high possibility to reduce the consumption wastes of electricity with ElectroGes autonomous energy management system.

REFERENCES

[1] ERSE-Entidade Reguladora dos Serviços Energéticos, “Preços de Referência no Mercado Liberalizado de Energia Elétrica e Gás Natural em Portugal Continental,” 2015.

[2] J. Amador, “PRODUÇÃO E CONSUMO DE ENERGIA EM PORTUGAL: Factos Estilizados,” *Bol. Económico | Banco Port.*, 2010.

[3] I. I. Pătru, M. Carabaş, M. Bărbulescu, and L. Gheorghe, “Smart home IoT system,” in *Networking in Education and Research: RoEduNet International Conference 15th Edition, RoEduNet 2016 - Proceedings*, 2016.

[4] F. K. Santoso and N. C. H. Vun, “Securing IoT for smart home system,” in *Proceedings of the International Symposium on Consumer Electronics, ISCE*, 2015.

[5] M. Mehta, “ESP8266 : A Breakthrough in Wireless Sensor Networks and Internet of Things,” *Int. J. Electron. Commun. Eng. Technol.*, 2015.

[6] OASIS (Organization for the Advancement of Structured Information Standards), “MQTT Version 3.1.1,” 2014.