Interfacing Two Energy Efficiency Solutions to Develop an Autonomous Load Control System for Low Voltage Installations

Torres-Tello Julio

Universidad de las Fuerzas Armadas ESPE, Departamento de Eléctrica y Electrónica, Sangolquí, Ecuador

Abstract: In this paper it is described an autonomous load control system for low voltage installations where we utilize on one side ABB's Ekip power controller, which uses a patented algorithm to decide whether a load in an electric system must be powered or not at a defined time instant; and on the other side the power limits that are obtained from the analyses of generation schemes and prices of electricity at a given moment, performed by Ventyx. The goal of this project is to develop a software interface between the information available at Ventyx about the electric energy market and ABB's Ekip controllers, to automatically and remotely configure the way in which they act over an electric installation. This work is mainly related to the Italian scenario, and an analysis of the Italian and European electric energy market is performed.

Keywords: Energy efficiency, programming, python, modbus, electric generation, low voltage.

Desarrollo de un Sistema Autónomo para el Control de Carga en Instalaciones de Bajo Voltaje por Medio de la Creación de un Interfaz Entre Dos Soluciones de Eficiencia Energética

Resumen: En este artículo se describe un sistema autónomo para el control de cargas en instalaciones de bajo voltaje; en donde utilizamos por un lado el controlador de carga Ekip de ABB, el cual usa un algoritmo patentado para decidir cuándo una carga debe ser conectada o desconectada de acuerdo a límites de potencia establecidos; y por otro lado, información generada por Ventyx que se obtiene del análisis de los esquemas de generación eléctrica y su precio en el mercado en un momento dado. El objetivo de este proyecto es desarrollar un interfaz en software entre la información disponible en Ventyx acerca de las condiciones del mercado eléctrico y los controladores Ekip de ABB a fin de poder configurar automáticamente y de forma remota la manera en que estos actúan sobre una instalación eléctrica. Este trabajo está enfocado principalmente al escenario italiano, por lo que se realiza también un análisis del mercado de la energía eléctrica en Europa y específicamente en Italia.

Palabras clave: Eficiencia energética, programación, python, modbus, generación eléctrica, bajo voltaje.

1. INTRODUCTION

It is possible to trace back the origins of the electric power systems to 1880's and since then it has been supporting the economic growth in Europe and the US, and other economies in more recent years. However, given that this technology has been so effective in satisfying all the industry needs for decades, it has stayed almost the same since its beginnings and most of the same basic principles are still applied (Casazza et al., 2003).

In recent years, the word *efficiency* is taking a stronger meaning in the electric energy field, given the new considerations that arise from many points of view such as environmental, economic, political, etc. Efficiency is defined as <<the ratio of the useful work performed by a machine or in a process to the total energy expended or heat taken in>> (Oxforddictionaries.com, 2015), and so it determines the quantity of resources that are being unused and that have the potential to be also transformed into electric energy in this case.

The goal of this project is to develop a software interface between the information available at Ventyx about the electric energy market and ABB's Ekip controllers, to automatically and remotely configure the way in which they act over an electric installation. The final product is a software tool developed in Python that makes it possible to configure multiple Ekip controllers at a time, remotely through an IP network.

In the first part of this paper we give a brief description of the electric energy market and analyze the reasons for the study of energy efficiency solutions; as well as the situation of the market in Europe and the characteristics that urge us to find a solution like the one here proposed.

jwtorres@espe.edu.ec

The following section describes the two energy efficiency solutions (one by ABB and one by Ventyx) for which we develop the interface; as well as the communications protocol over what we base our development.

Finally we describe the developed interface, how it has been developed and how it works; and we compare a scenario with and without the proposed system.

2. SITUATION OF THE ELECTRIC ENERGY MARKET

Energy efficiency solutions like the one described in this paper try to tackle the problem generated by the fact that the world keeps increasing its energy consumption as we can see in the International Energy Agency Statistics graphics (Figure 1).

However, consumption is not the only problem. Until the third quarter of 2014 oil prices were very high (more than 100 USD per barrel) and it is always a volatile market, as seen in Figure 2. This situation urges us to find alternatives to the dependency on petroleum and coal, but also to make electric systems more efficient. There are campaigns focused on the users, trying to push them to use electricity in a better way, but that is not the only possibility. Large scale solutions are also required. <<On average power plants consume up to 7\% of the electricity they generate, while industrial sites account for around 33\% of global energy use and buildings account for nearly 40\% of energy consumed. These figures could be cut by 10\% to 30\% by optimizing the various processes and systems that run the plants.>> (Ferrara, 2013).

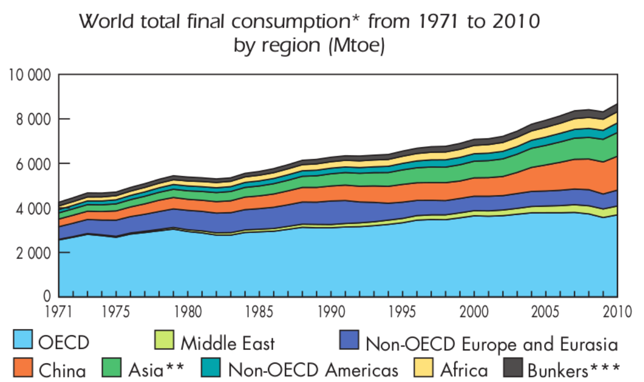

Figure 1. World total energy consumption. (International Energy Agency, 2012).

One way to optimize those processes and systems is by means of virtual power plants and smart grids. A *Virtual Power Plant (VPP)* is an arrangement of distributed generation installations, where the electric power can be generated by different methods, such as wind, solar, traditional means, etc., that is managed by a centralized control entity. And a *Smart Grid* is defined as a technique which <<combing time-based prices with the technologies that can be set by users to automatically control their use and self-production, lowering their power costs and offering other benefits such as increased reliability to the system as a whole.>> (Fox-Penner, 2010).

From these definitions it is clear that in countries like Italy, where this information is available to anyone (Figure 3), prices can be used to motivate the customers to keep their power use below certain margins, preventing the whole system to exceed the installed capacity. In the end, the smart grid will give customers much more control over their own power use and make dynamic pricing a universal condition in electric systems. However if we give the user this information and he/she has to make decisions about which device and when it has to be used, it becomes a tedious task that most people would refuse to do. Therefore there is the necessity of an autonomous system that can use the information about electricity prices to set limits to the consumption in an installation.

In this work we consider that ABB has developed the Ekip power controller, a device that can connect and disconnect electric loads which will be later described; and that Ventyx (an ABB company) has a system that analyzes the electric market and generates power limits for a given user. The goal of this work is to develop an interface between these two; in order to create an autonomous system that can get information about the electric market and set the values in the load controller; in order to use the electricity in such a way that loads are connected when energy is cheaper and disconnected when it is expensive (with some considerations about which ones and when they can be unpowered). This interface was named Ekip Management Software (EMS).

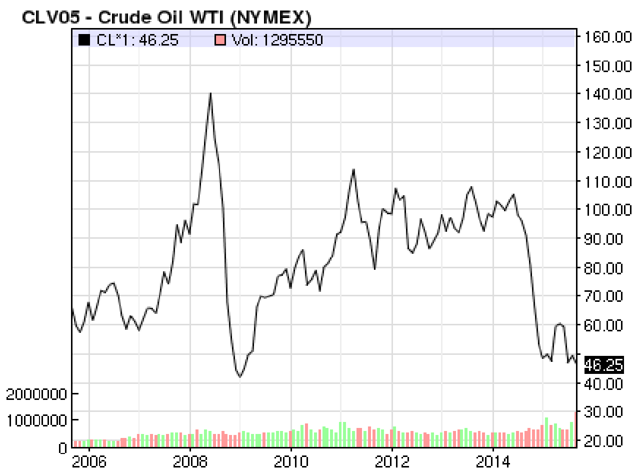

Figure 2. Crude oil prices in dollars during the last 10 years. (Nasdaq.com, 2015).

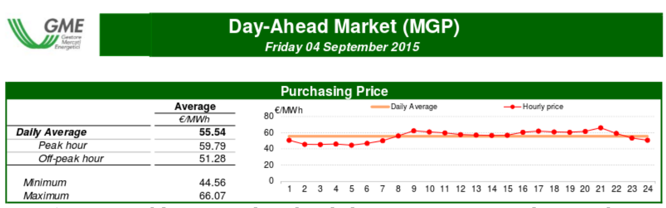

Figure 3. Public access day-ahead electricity prices in Italy given by "Gestore del Mercati Energetici". (Mercatoelettrico.org, 2015).

2.1. The Italian and European Situation

In the Italian case there was a process called *deregulation* or *liberalization* that took place more than ten years ago and was similar to other processes happening all around Europe. As a result, this deregulation divided the electric system into four separate actors: Generation, Transmission, Distribution and Retail. This concept is widely applied in Europe consenting small variations in each specific case; and especially for the last two aspects, there are some variations in the models in countries such as Germany and Turkey.

In the case of this work, there is more interest in the last two parts of this market model (distribution and retail) because in here consumption forecast is pursued. The involved companies have to predict the future demand of electric energy from its users; and it has to be done for a variety of time periods, going from months (3, 6, 12) down to seconds.

What is done here is that the retailer creates a price band with the expected maximum and minimum values that will be needed during a certain period (as long as possible), in order to satisfy its customers demand, using this information to buy energy from the distribution company at a good price.

Later, this forecast will be adjusted and the retailer will buy (or sell if possible) more energy if needed in week-ahead, day-ahead plans, or even in shorter times. The later has to be avoided as much as possible since in those cases the energy is usually much more expensive; and that is why a precise forecast is desirable. Those last-minute acquisitions are much more expensive, mainly because of the generators limitations: A power plant cannot be turn on and off easily in such short times.

In both cases (retail and distribution), besides a precise consumption forecast, there is another solution to avoid these unexpected, expensive, last-minute energy purchases, which is cutting *(or shaving)* the consumption peaks. It is important to note that the rates at which the customers use the electric power is not usually measured instantaneously, but are calculated as an average level of usage during certain amounts of time; being 15, 30, and 60 minutes the most common used time intervals (Casazza et al., 2003).

There are studies (Amistadi, n.d.) showing how this peak-shaving strategy can affect the final performance of the electric power system as a whole. For example, the top 6\% of the electric capacity in France is used for only 1\% of the time during one year, as a clear display of an overestimation of the system that could be avoided by means of a peak-shaving strategy. Also, the same studies show that customers could pay less if they can use this technology; for example, there is an average 23\% reduction in pilots conducted in the US.

3. THE ENERGY EFFICIENCY SOLUTIONS

3.1. The Ekip power controller

ABB is a world leading company in power and automation technologies, based in Switzerland, that have developed the Ekip power controller, which is an absorbed-power based controller that uses a *patented algorithm* to decide if a given load must be powered or not at a defined time instant. The Ekip controller needs one circuit breaker (used as overcurrent protection) that has the capability of acting over a number of electric loads by connecting or disconnecting one or many of them at a given time.

In order to accomplish its tasks, the Ekip controller must constantly measure the energy absorbed by the whole installation, and with this information (energy/power) and the proper settings that the user has introduced in the system configuration, it will be able to connect or disconnect a given load (Load management with Ekip power controller for SACE Emax 2, 2012). It is important to mention that the controller will not only manage the passive loads that are consuming energy, but it can also connect the reserve generation device, that will be understood as a *negative* load.

To obtain the information about the consumed energy, the controller uses the voltage and current measurements that the trip unit must keep in track. As for the other part of the required information, the content given by the user, consisting mainly in a few default power limits, can be directly set on the device through the user interface that it has or through a computer using USB, serial or Ethernet ports; and it is in the later where we operated.

3.1.1. The load control algorithm

This algorithm is based in the following four-step procedure (Load management with Ekip power controller for SACE Emax 2, 2012):

1. *Measurement.* The controller measures the total power in the system and integrates this power in one time period, usually 15 to 30 minutes. This measurement is reset to zero every time interval, so that one can obtain the energy consumed for each one of this time slots.

2. *Synchronization.* To maintain an accurate control of the system according to the information provided.

3. *Evaluation.* The algorithm calculates the current scenario and predicts whether the total mean power will overcome the established limits or not. This prediction is done by dividing the plane <t,E> (energy versus time) into three regions or scenarios limited by the curves Emax and Emin, as shown in Figure 4.

If the instant power through the circuit breaker falls within the two lines (limits) it maintains the actual load configuration. If it falls below the curve Emin, the decision will be to increase the number of loads that are connected, or if it locates over the curve Emax, the decision will be to decrease the number of loads (Load management with Ekip power controller for SACE Emax 2, 2012).

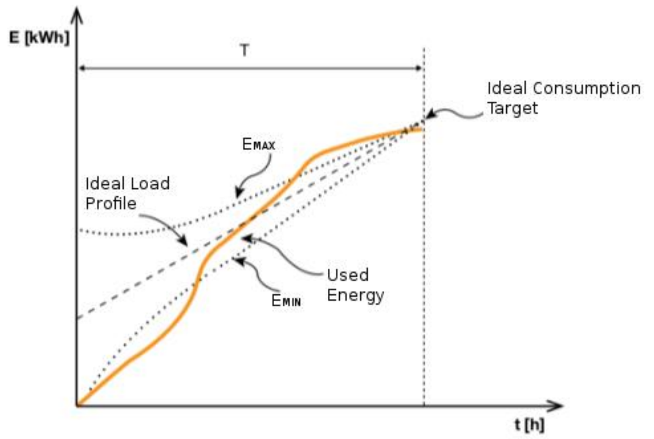

Figure 4. Energy consumption curve and its maximum and minimum limits. (Load management with Ekip power controller for SACE Emax 2, 2012).

4. *Load Management.* After evaluation and as a result of that step, this process involves another level of decision, in which the loads are connected or disconnected according to three criteria, which are: Priority, Respect times and Reordering. These parameters are useful to decide which one of the loads is going to be connected or disconnected at a given time interval.

3.2. Power limits generated by Ventyx

In order to work properly, the proposed system needs the power limits information; and this is generated by Ventyx for each customer. Ventyx performs a market analysis from the electricity prices (Figure 3) and based on this information and the specific requirements of the customers it can calculate in real time the power limits that should be used by the installation in order to save as much money as possible. As a result Ventyx produces XML files with this information; and these files are the ones that we use in the proposed system.

3.3. Modbus Protocol, the base of our interface

As officially defined, <<MODBUS is an application layer messaging protocol, positioned at level 7 of the OSI model, which provides client/server communication between devices connected on different types of buses or networks.>> (Modbus application protocol specification, 2012).

The Modbus standard is based in a series of function codes that allow the devices to write and/or read registers or bits in the connected devices, for both serial and TCP implementations. The general Modbus protocol defines a simple protocol data unit (PDU) independent of the layers below. This PDU (Figure 5) contains an 'Additional address' and an 'Error check' fields, used for serial implementations, which are outside the scope of this work, given that these functionalities are already implemented by the underlying protocols in the case of the TCP implementation. As for Modbus TCP, the defined fields are:

- *Data.* Is the payload itself, containing the information that will be read or written into the internal registers of the device in a big-endian representation.

- *Function Codes (FC).* Characterize the type of data that will be handled and the kind of operation that will take place upon them.

The whole communication system is designed over the Modbus TCP standard, and one software tool that was relevant for the development of EMS, was Pymodbus which << is a full Modbus asynchronous communications core. It can also be used without any third party dependencies (aside from pyserial) for both reading and writing the registers of interest>> (Pymodbus project, 2011).

Here it is not used the asynchronous communication mode, since the EMS will only operate via IP networks. Beside this, Pymodbus implements the Modbus protocol as it is, but in order to be compatible with the Ekip implementation, it had to be adapted to it; as later described.

4. THE DEVELOPED INTERFACE

4.1. The Ekip Management Software (EMS)

The final product of this work is a software implementation developed in Python that makes it possible to configure multiple Ekip controllers at a time, remotely by means of an IP network. As seen in Figure 6, on one side we have from Ventyx system, the power limits, nodes identification, timestamps; and on the other side, the parameters that are directly set by the administrator of the EMS into the system.

Finally, once all this information is available, the EMS tool can process it and generates the corresponding configuration commands that will remotely program the controllers through an IP network. Python has been chosen since it is nowadays a revolutionary tool within the world of academics and applied research. (Herrera, 2013).

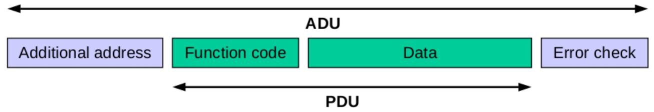

Figure 5. The Modbus protocol frame, as defined in the standard. (Modbus application protocol specification, 2012).

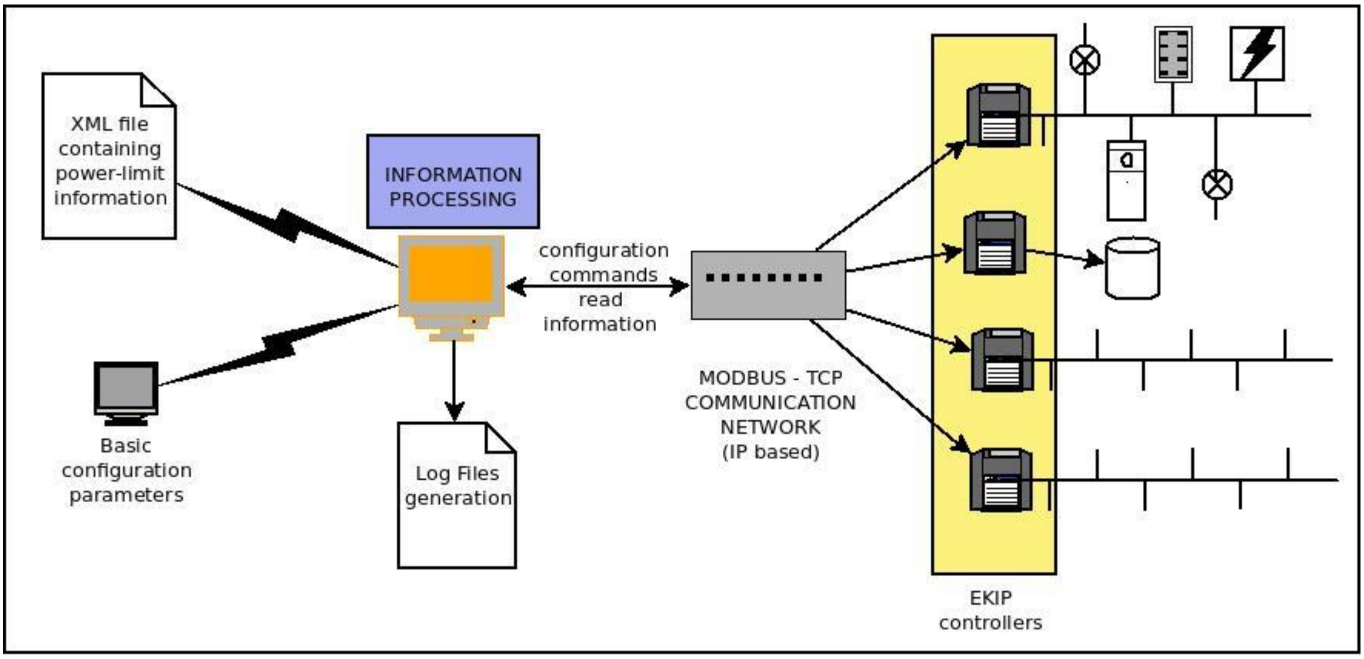

Figure 6. General working process of the EMS tool.

Another important part of the EMS tool, besides writing the configuration information, is that it will constantly read some registers inside the controllers, in order to generate log files.

EMS has been designed in order to accomplish two main tasks: an automatic configuration process and the generation of log files; tasks that have produced two independent modules that work together inside the primary program. In Figure 7 it is also indicated that the program can go back to the beginning from any of the two modules; this situation happens when a new XML input file is found. EMS is constantly checking for new information; and when it arrives, it starts over the configuration/log generation process, according to the settings in the configuration file. Once the EMS software has been started, it can keep running without the need for any kind of intervention, as long as it has new input files, at least once a day.

The Ekip Controller Automatic Configuration Tool, here called ECACT module, has the main goal of retrieving data from XML files that contain information about the maximum powers for the different nodes (Ekip controllers) in a time based scheme; in order to automatically configure those nodes in an iterative process, assigning the maximum powers to the nodes, and updating the information in the controllers for every time slot (which has been defined in fifteen minutes, according to the information generated by Ventyx).

Once this information from the input XML file has been extracted and processed (stored, standardized, organized and matched), the ECACT module generates configuration commands that are periodically sent to all the involved nodes, in order to keep their power limits set to the values stated in the input XML file.

The second module, called ECLGT, which stands for Ekip Controller Logs Generation Tool; as its name indicates is the one in charge of the generation of log files for the nodes (Ekip controllers) of interest. The nodes for which the log files are generated are defined by their IP addresses, being these either directly gotten from the input XML file (if only this module is being used), or obtained from the ECACT module in case that it has been used first. The log files are updated with new information about the status of all the involved nodes every minute, and stored every 15 and/or 60 minutes, depending on the kind of log files that are being generated.

The EMS tool has been created as a python script that after installing the corresponding dependencies, could be directly executed in Linux-based systems. However it has also been ported to Windows systems by means of the generation of an executable file (.exe).

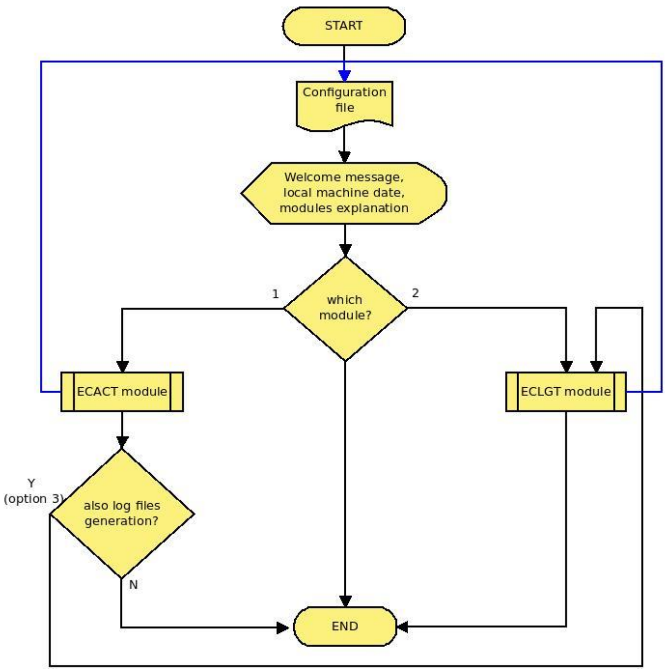

Figure 7. General EMS workflow diagram.

4.2. Modbus Implementation in the EMS tool

The implementation of the Modbus protocol in the EMS tool consists basically in adapting the function codes to their usage in the Ekip controller, considering only the ones that perform the operations that access the required registers. The implementation of Modbus protocol in the Ekip controller uses only registers (not independent bits), and they can be accessed for both read and write; although most of them can only be read or written, depending on the information they store.

Almost no modifications are needed to the standard implementation, except for the default unit identifier. On TCP/IP, the MODBUS server is addressed using its IP address; therefore, the MODBUS Unit Identifier is useless. The value 0xFF has to be used (Modbus messaging on TCP/IP implementation guide, 2006). Accordingly the value implemented in the Pymodbus library is set to '0' (0x00) by default, which means that all connected devices should respond to a request; but due to the adaptation of the standard that was done by ABB in the Ekip controllers, it had to be changed to '1' (0x01). For more information about these fields you are referred to Modbus application protocol specification (2012) and Modbus messaging on TCP/IP implementation guide (2006).

5. TESTS AND RESULTS

Tests were initially done by using a Modbus TCP simulation software, in order to understand the basic communication process of the developed software with a Modbus capable device and how it responds with the modified preset values, in a generic device.

The second step was to perform the tests with a real Ekip controller. At that stage, a packet analyzer was another useful tool, in order to have a complete understanding of the commands that were being both sent and received between the testing workstation and the controller. It was especially useful in order to 'tune' the communications scheme.

Finally, it was needed to prove that the configurations were being loaded and correctly used by the Ekip controller; and to perform this task we used a software called Ekip Connect, developed by ABB that can link to the Ekip controller via serial port or USB (the one that was used for the tests) and display the configuration information.

Once the software was tested for its functionalities, it was also validated in a controlled environment in ABB labs in Bergamo, Italy; where it showed that the desired results were achieved, which means that EMS could extract the power limits information from XML files and configure Ekip controllers over an IP network with a minimal interaction from the user (Figure 8); obtaining a configuration much closer to the information available from the electric market analysis as it can be seen in Figure 9, where we can observe that using the EMS tool does not necessarily mean that the power limits are smaller, but that these are closer to the information that Ventyx can generate from the electric market analysis. Given the Ventyx model which generates higher power limits when electricity is cheaper, in the end it represents using the maximum amount of electric energy at the lowest possible price.

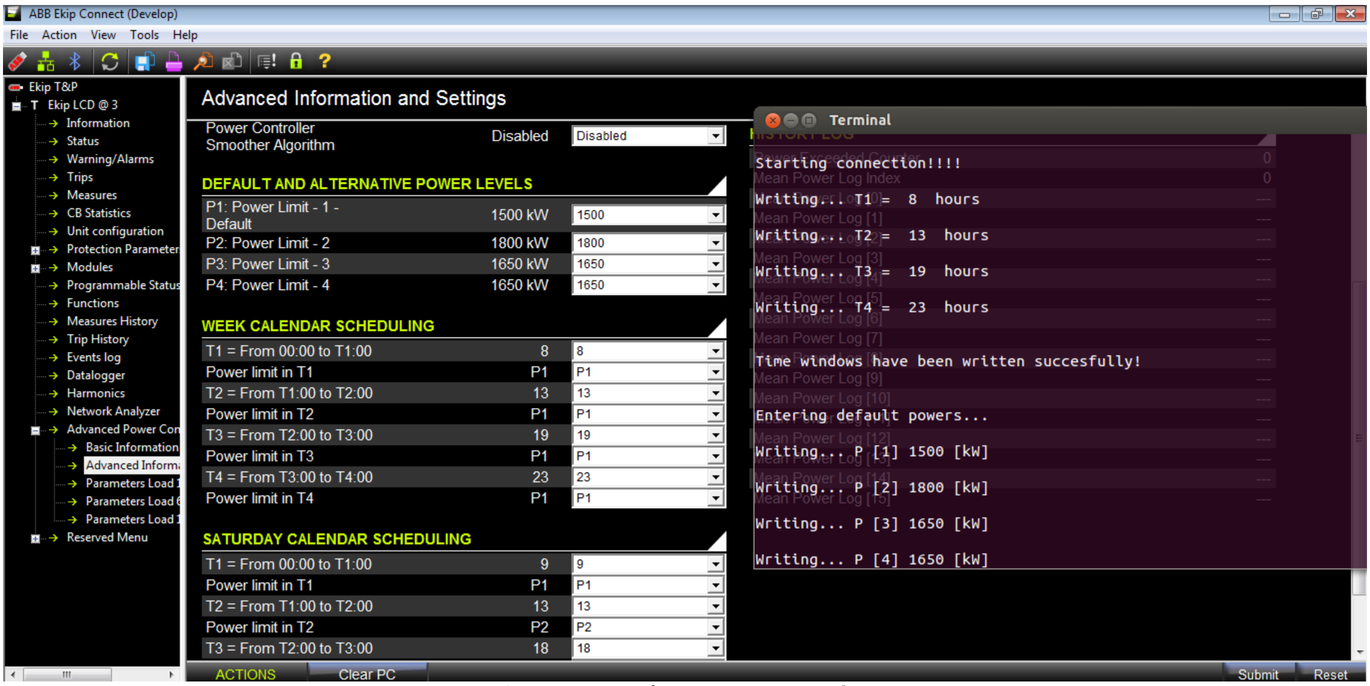

Figure 8. EMS software running screenshot.

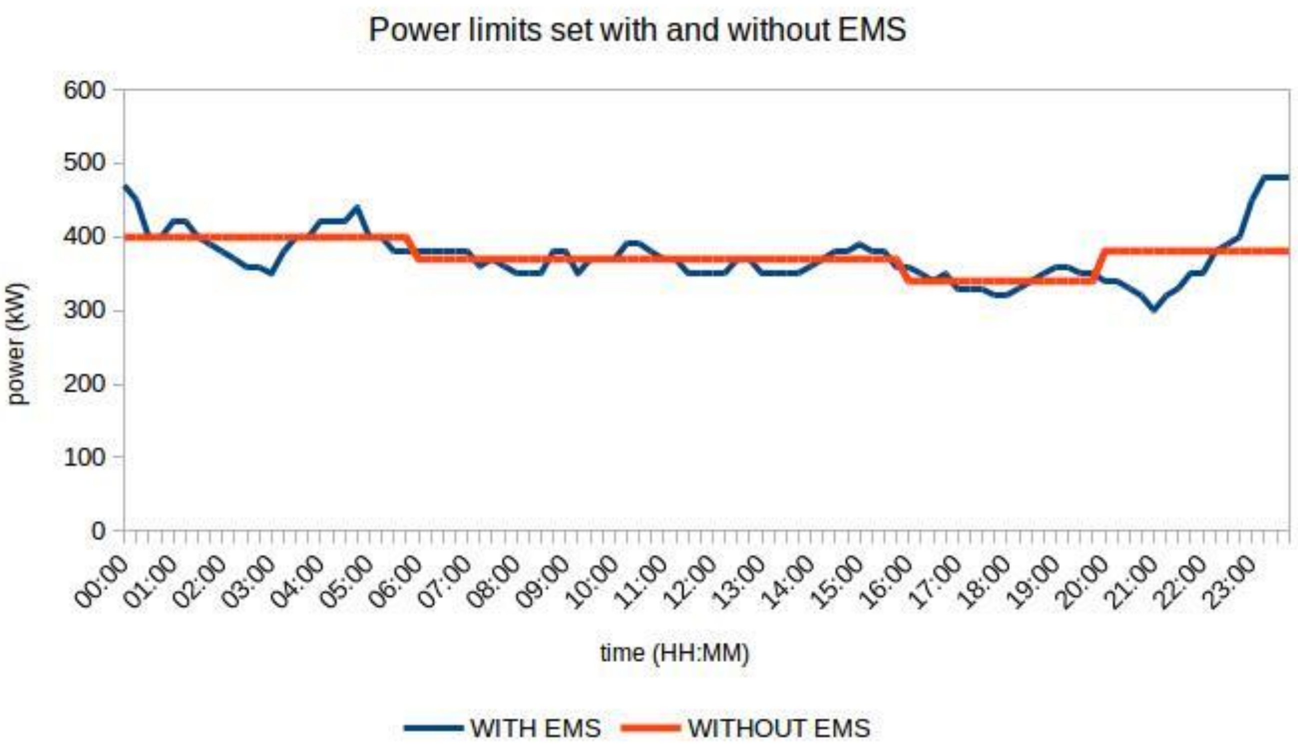

Figure 9. Configuration profile of the power limits in the EMS controller with and without using the EMS tool.

By using the technique mentioned in the operation of the ECACT module, compared to the original capabilities of the Ekip power controller, a higher level of granularity is achieved by fitting in a closer way the power limits profile that has been defined by the company (Figure 9). However, if there is no information available for each fifteen-minute time slot along the whole day; there are default power limits (included in the XML input file) that would be used to fill those spaces.

Another result is the implementation of the Modbus protocol in Python by means of the Pymodbus tool, with some specific customizations related to the usage of this protocol in the Ekip controller, as described in section 4.2. This means that Pymodbus has been found to be a very flexible tool.

6. CONCLUSIONS

The Ekip power controllers define only four time ranges with their correspondent power limits and these values stay fixed in time (with only three possible different schemes per week); which was changed by the adoption of EMS. As a result this software tool makes it possible to have a more precise management of the electric energy consumption. Given that the power limits are obtained considering whether the electricity prices are low or not, if we can accurately track this information and use it to set the corresponding values in the Ekip controller, we can conclude that this would represent a more efficient usage of the resources and savings in the electric bills (the maximum possible amount of electric energy at the lowest possible price), when implementing this system in a real situation.

Another benefit that results from using the EMS tool is that the user is released from the obligation of keeping a constant track of the information about the possible power limits for the installation and also, since it provides automatic configuration of some values, the user will have a smaller level of interaction with the controller or controllers.

With reference to a future work, the first task that should be done is finding out the precise amounts of electric energy and/or money that could be saved by using the EMS tool. In order to do this the system (hardware and software) should be installed in an industrial plant, hotel building, shopping mall, etc. which is the target market of the Ekip power controller; and the electric energy consumption compared when using and not using the proposed solution. It might also be necessary a statistical analysis of the results, which would provide a better idea of the behavior of the system.

It is also important to analyze the specific situation of the electric market in other countries, to determine if the required information needed for this system to work is available and how it could be used.

Studying the security issues that could come from the fact that all the configuration process happens over an IP network (which could even have public access) is also recommended.

ACKNOWLEDGMENT

This paper is based on the results of the Master's thesis "Design and Development of a Software Tool for the Management of ABB's Ekip Controllers" in Electronic Engineering presented at Politecnico di Milano (Italy) in 2013, which was developed under the guidance of Prof.

Enrico Ragaini, member of the Department of Electronics, Information and Bioengineering

REFERENCES

Amistadi, E. *An off-the-shell solution for virtual power plants*, unpublished.

Casazza, J., and Delea, F. (2003). *Understanding electric power systems: An overview of the technology and the marketplace*. Hoboken, NJ: John Wiley & Sons, Inc.

Ferrara,F. (2013). *Energy efficiency as a key enabler. ABB Conversations.* Available: http://www.abb-conversations.com/2013/07/energy-efficiency-as-a-key-enabler/. (June, 2015).

Fox-Penner, P. (2010). *Smart Power: Climate change, the smart grid, and the future of electric utilities*. Washington, DC: Island Press.

Herrera, R. (2013). Herramientas de software libre para aplicaciones en ciencias e ingeniería. *Revista Politécnica*. 32, 4.

International Energy Agency. (2012). *Key World Energy Statistics*. Paris, France: OECD/IEA.

Load management with Ekip power controller for SACE Emax 2. ABB SACE. (2012). Bergamo, Italy.

Mercatoelettrico.org. (2015) *GME - Gestore dei Mercati Energetici SpA*. Available: http://www.mercatoelettrico.org/En/Default.aspx. (September, 2015).

Modbus application protocol specification, V1.1b3. The Modbus Organization. (2012). Hopkinton, MA.

Modbus messaging on TCP/IP implementation guide, v 1.0b. The Modbus Organization. (2006). Hopkinton, MA.

Nasdaq.com. (2015) *Commodities: Latest Crude Oil Price Chart.* Available: http://www.nasdaq.com/markets/crude-oil.aspx?timeframe=10y. (September, 2015).

Oxforddictionaries.com, 'efficiency - definition of efficiency in English from the Oxford dictionary', 2015. Available: http://www.oxforddictionaries.com/definition/english/efficiency. (September, 2015).

Pymodbus project. (2011). *Pymodbus version 0.9*. Available: https://code.google.com/p/pymodbus.