

# Design and Development of a Software Tool for the Management Of ABB's Ekip Controllers

Torres Tello J.\*

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

e-mail: jwrtorres@espe.edu.ec/phone:+593995435105

---

**Resumen:** La principal motivación para la realización de este proyecto es la creciente demanda de energía eléctrica a nivel mundial y la necesidad de desarrollar estrategias que permitan una generación y uso más eficiente de la misma. Uno de los componentes que ayuda en esta tarea es el controlador de carga Ekip de ABB, el cual usa un algoritmo patentado para decidir cuando una carga debe ser conectada o desconectada de acuerdo a límites de potencia establecidos y que se obtienen de análisis de los esquemas de generación eléctrica y su precio en el mercado en un momento dado. El objetivo de este proyecto es el de desarrollar un interfaz entre la información disponible en ABB a cerca de las condiciones del mercado eléctrico y los controladores Ekip; a fin de poder configurar automáticamente y de forma remota la manera en que estos actúan sobre una instalación eléctrica. Dicho interfaz es el programa informático denominado EMS, herramienta desarrollada en Python que recibe la información a través de archivos XML, procesa la información y configura los controladores a través de una red IP; además de leer la información del estado actual de los controladores para generar archivos de registro. Este estudio está enfocado principalmente al escenario italiano, pero con algunas consideraciones podría aplicarse a la situación de otros países.

**Palabras claves:** Eficiencia energética; programación; python; modbus; redes; generación eléctrica; cargas eléctricas; ekip; control.

**Abstract:** The main motivation for the realization of this project is the worldwide growing electric energy demand and the necessity of developing strategies that provide more efficient generation and use schemes of it. One out of the many components that might make it possible is 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, according to power limits that are obtained from the analyses of generation schemes and prices of electricity at a given moment. The goal of this project is to develop an interface between the information available at ABB about the electric energy market and the Ekip controllers, to automatically and remotely configure the way in which they act over an electric installation; providing a power limit profile. In order to achieve this the EMS tool has been developed, which is a program written in Python that fetches the information from XML files, processes the data and configures the controllers over an IP network; and also reads information about the status of the controllers in order to generate log files. This study is mainly related to the Italian scenario, but with some considerations it could be applied the situation of other countries.

**Keywords:** Energy efficiency; programming; python; modbus; networking; electric generation; electric loads; ekip; control.

---

## 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 hasn't changed so much since its beginnings and most of the same basic principles are still applied now.

---

Manuscript received June 11, 2015.

This paper is based on the results of the homonym Master's thesis 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.

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, economical, political, etc. Efficiency is often defined as the ratio of a system's output to its input,

and so it determines the quantity of resources that are being unused and that have the potential to be also transformed into the mentioned output, or electric energy in this case. In electric systems, this input (the fuel), has been not a limitation because it has always been available, inexpensive and environmental issues haven't been really considered.

## 2. SITUATION OF THE ELECTRIC ENERGY MARKET

### 2.1 Energy consumption

The main problem here arises from the fact that the world keeps increasing its energy consumption no matter what, as we can see in the International Energy Agency Statistics graphics (Fig. 1).

One way to reduce and prevent the damaging effects caused by the electric generation is by reducing the consumption of energy that is produced by burning fuel, using environmentally friendly energy generation methods instead (most renewable energies), but also by using the available energy in the most efficient way.

### 2.2 Economic Concerns

Just a few months ago the oil prices were very high (more than 100 USD per barrel) and it's always a volatile market. This situation makes urgent the need for finding alternatives to the dependency on petroleum and coal, but also the need of making electric systems more efficient. If energy becomes so costly, it is not profitable to keep wasting it at all levels. There are campaigns focused on the users, trying to push them to use electricity in a better way, but that is not the only solution. Large scale solutions are also required.

### 2.3 Smart Grids and Virtual Power Plants

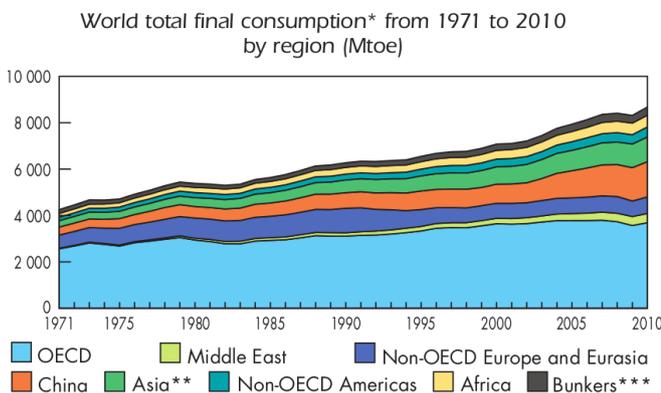


Figure 1. World total energy consumption.[6]

«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.» [4] Some concepts that are well related to what is called smart power, are 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 «combining 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.» [5]

From these definitions it is concluded that 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.

### 2.4 The Market

In the Italian case there was a process called *deregulation* or *liberalization* that took place more than 10 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.

#### 2.4.1 Consumption forecast

Given this market model, consumption forecast is performed in order to save money, since the energy prices are not fixed, and they vary according to a wide range of factors, including the geopolitical situation, the weather, reliability of the systems, seasonality, and of course demand. In the case of this project, there is more interest in the last two parts of this market model (distribution and retail) because in here consumption forecast is pursued. The companies here involved 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 last 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.

#### 2.4.2 Consumption peaks

In both cases (retail and distribution), besides a precise consumption forecast, there is another solution to avoid these unexpected, expensive, last-minute energy purchases, and this is done by not requiring them. And one especial strategy is to cut (or shave) the consumption peaks. Here, it should be taken into account the fact 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. [3] There are studies [2] 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.

#### 2.5 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. 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 communication or by means of serial or Ethernet ports.

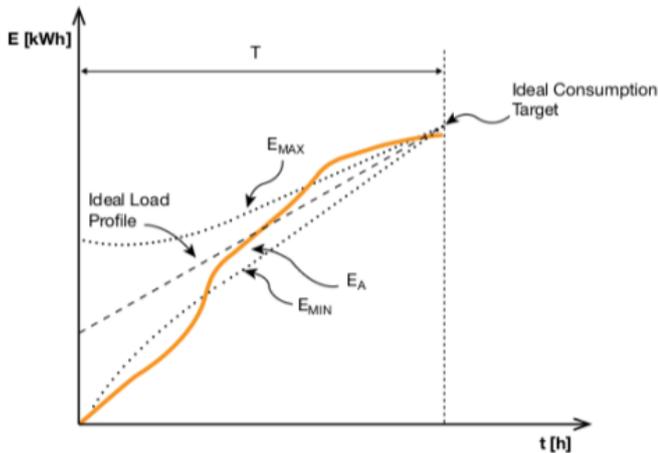
##### 2.5.1 The load control algorithm

This algorithm is based in the following four-step procedure:

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 most important step in this procedure is the evaluation of the scenario in which the controller operates at a given time instant and the possible actions that it could perform. 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  $\langle t, E \rangle$  (energy versus time) into three regions or scenarios limited by the curves  $E_{max}$  and  $E_{min}$ , as shown in the figure 2.

The calculation is performed by measuring the instant power through the circuit breaker and obtaining the region in the plane where the power (or energy) consumption is located, if it falls within the two lines (limits) it maintains the actual load configuration. If it falls remarkably below the curve  $E_{min}$ , the decision will be to increase the number of loads that are connected, or if it locates over the curve  $E_{max}$ , the decision will be to decrease the number of loads. [1]

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.



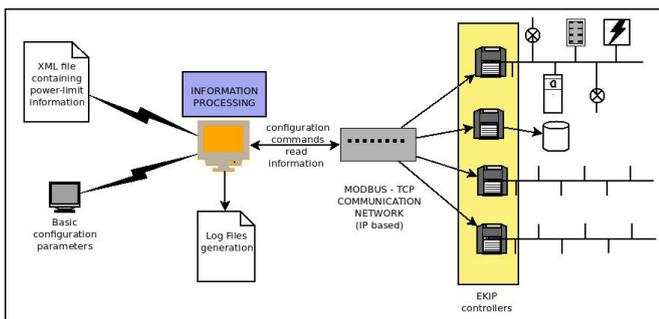
**Figure 2.** Energy consumption curve and its maximum and minimum limits.[1]

### 3. THE PROPOSED ENERGY EFFICIENCY SCHEME

One of the characteristics of the Ekip controller is that it can be remotely configured using an Ethernet connection, which means that communication networks can be used to configure multiple Ekip controllers connected to a unique central device that could manage some or all of the functions of the controller. In this work, using the information about energy prices that the market analyses provide, power limits defined in a day-ahead scheme are being automatically loaded into the controllers through an IP network.

#### 3.1 The Ekip Management Software

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. It becomes a precise method of saving electric energy and money in one or more low voltage installations.



**Figure 3.** General working process of the EMS tool.

As seen in the figure 3, there are two kinds of inputs to the system; on one side there is the input information that the EMS will have from an external system (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's initial configuration settings. Finally, once all this information is available, the EMS tool can process it and generate the corresponding configuration commands that will remotely program the controllers through an IP network. Since the IP addresses can be either static or dynamic (provided by a DHCP server), these must be previously known by the system that generates the configuration information, and they will be included in the input XML file.

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. To do this, EMS uses some software tools (related to Python), and its communication system is designed over the Modbus TCP standard.

#### 3.1.1 Software Tools

Python has some external libraries that have been helpful in order to implement the functions of the EMS tool. These libraries are:

- **Pymodbus** as described in the protocol implementation «(it) is a full Modbus asynchronous communications core. It can also be used without any third party dependencies (aside from pycserial) for both reading and writing the registers of interest.» [8] Here it is not used the asynchronous communication mode, since the EMS will only operate via IP networks.
- **LXML** is a Python library that provides an easy to use processing of XML and HTML files. It is needed to process the files containing the power limits information by analyzing the XML tags and obtaining the stored information.
- **XLWT** is a library used to create spreadsheet files that are useful in order to generate the log files. The spreadsheets have a structure that cannot be natively handled by Python language.

All the development of the software, including the installation of these libraries, the programming of the code for the EMS tool, and all the tests were done under

GNU/Linux environment (Ubuntu 13.04). However, the final tests in ABB laboratories were done in a Microsoft Windows environment, generating the EXE file by using the tool py2exe.

### 3.2 The Modbus Protocol

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.» [7]

The Modbus standard is based in a series of function codes that allow the devices to write and/or read registers 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 contains an 'Additional address' and an 'Error check' field, 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):** The second field of the PDU corresponds to the function codes which characterize the type of data that will be handled and the kind of operation that will take place upon them.

#### 3.2.1 Implementation in the EMS tool

The implementation of the Modbus protocol in the EMS tool consists basically in the adaptation of the function codes to their usage in the Ekip controller, considering only the ones that perform the operations that access the required registers. Almost no modifications are needed to the standard implementation, except for the default unit identifier value implemented in the Pymodbus library that is set to '0' (0x00) by default, which means that all connected devices should respond to a request; but due to the implementation of the standard that was done by ABB in the Ekip controllers, it had to be changed to '1' (0x01).

### 3.3 Main Workflow of EMS

The Ekip Management Software 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 the figure 4 it is

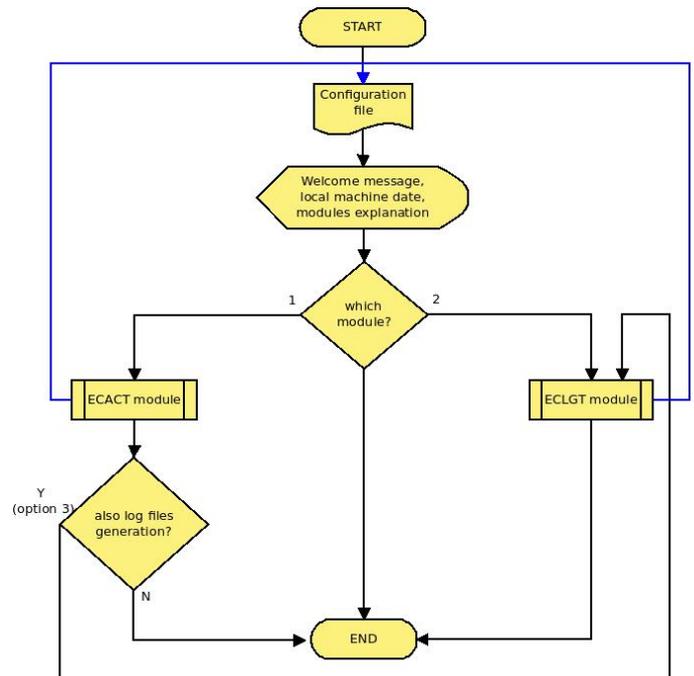


Figure 4. General EMS workflow diagram.

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.

#### 3.3.1 Configuration

There is a text file (config.txt) that contains the basic initial information that EMS will use in order to operate. This file has been created in order to make the software tool as independent from the user as possible. There are five parameters specified in this file, which are:

- The directory where to look for new input XML files.
- The directory where to save the old input XML files.
- Which module (EACT, ECLGT, or both) will be used.
- The directory where the log files are going to be stored.
- The format of those log files (XML and/or spreadsheet).

These parameters can be modified only before the program starts or before a new XML input file arrives. Otherwise, if there is a need for changing the parameters, the EMS tool has to be restarted.

### 3.3.2 The ECACT Module

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 available in the XML files).

Once this information from the input XML file has been extracted and processed (stored, standardized, organized and matched), the ECACT module will generate configuration commands that will be sent periodically to all the involved nodes, in order to keep their power limits set to the values stated in the input XML file. Then the program waits for a new XML file and in case of finding it, it will start over the whole process; otherwise it just waits for other fifteen minutes.

By using this technique, 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. 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.

### 3.3.3 The ECLGT Module

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 will be generated are defined by their IP addresses, being these either directly gotten from the input XML file (in case that 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.

As for the collected data, these are:

- Measurement time, which is the timestamp taken from the controller itself.
- Active Power Total, measured in watts (W).

- Active Energy Total, measured in kilowatts-hour (kWh).
- Default Power Limit, measured in kilowatts (kW).
- Evaluation Window, measured in minutes (min). By default it is used the value of fifteen.
- Elapsed Time, measured in minutes (min). It refers to the amount of time that has passed since the evaluation window started.
- Mean Power, measured in kilowatts (kW).
- Energy Log Index, that indicates which one of the following indexes is being used at a certain time.
- Energy logs (0 to 15), measured in kilowatts (kW). They correspond to the mean power that has been measured during a time window; and they take place sequentially.

Due to scalability and flexibility reasons, both of the modules previously described make use of functions, in fact the modules themselves are also defined as functions as a part of the main tool.

### 3.4 Utilization of the EMS software

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). After running or executing the corresponding file, the user doesn't have to interact with the program because it will keep running independently as far as it is fed with new XML input files at least once a day. However, the EMS tool requires a configuration file (plain text) that has to be stored in a specific location, and that will follow the specified format.

## 4. TESTS AND RESULTS

The tests were initially done by using a Modbus TCP simulator software, in order to understand the basic communication process with a Modbus capable device, how to read and write registers and modifying the 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 workstation and the controller. It was specially 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 there is a software called Ekip Connect, developed by ABB that

can link to the Ekip controller via serial port or USB (the last one was used for 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 it achieved the desired results, 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; obtaining a configuration much closer to the information available from the electric market analysis, which would be translated into a more energy efficient scheme.

## 5. CONCLUSIONS

The Ekip power controllers define only four time ranges with their correspondent power limits and these values are kept fixed in time (with only three 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, which represents a more efficient usage of the resources and savings in the electric bills.

Another benefit that results from using the EMS tool is that the user doesn't have to worry about 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 the determination of the precise amounts of electric energy and money that could be saved by using the EMS tool. In the case of the logs generation, it would be useful to determine if the fields that have been included are actually useful and it might be necessary also a statistical analysis of the results, which would provide a better idea of the behavior of the system. It would also be interesting to study 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).

## REFERENCES

- [1] ABB SACE, "Load management with Ekip power controller for SACE Emax 2", Bergamo, Italy: ABB, 2012, pp. 6-8.
- [2] E. Amistadi, "An off-the-shell solution for virtual power plants", Ventyx, an ABB company, Milan, Italy, Mar. 2013.

- [3] J. Casazza & F. Delea, "Understanding electric power systems: An overview of the technology and the marketplace", Washington, DC, U.S.: John Wiley & Sons, Inc., 2003, pp. 44.
- [4] F. Ferrara, "Energy efficiency as a key enabler", ABB Conversations, Jul. 2013. [Online] Available: <http://www.abb-conversations.com/2013/07/energy-efficiency-as-a-key-enabler/> [Accessed: Jun. 05, 2015]
- [5] P. Fox-Penner, "Smart Power: Climate change, the smart grid, and the future of electric utilities". Washington, DC, U.S.: Island Press, 2012, pp. 34.
- [6] International Energy Agency (IEA), "Key World Energy Statistics" (OECD/IEA), Paris, France: Soregraph, 2012, pp. 32.
- [7] The Modbus Organization, "Modbus application protocol specification V1.1b3", 2012, pp. 2.
- [8] Pymodbus (Version 0.9). [Software]. Pymodbus project, 2011.