

# **Advanced Architectures and Control Concepts for MORE MICROGRIDS**

**Contract No: PL019864**

**WORK PACKAGE A**

**DA6. Load Controller with Plug n' Play Capabilities**

**Final Version**

**October 2009**

**Document Information**

**Title:** Load Controller with Plug n' Play Capabilities

**Date:** 20-10-2009

**Task(s):**

<b>Coordination:</b>	Aris Dimeas	adimeas@power.ece.ntua.gr
<b>Authors:</b>	Nikos Hatziargyriou	nh@power.ece.ntua.gr
	Aris Dimeas	adimeas@power.ece.ntua.gr
	Fotis Psomadelis	dea@anco.gr

**Access:**

<input type="checkbox"/>	<b>Project Consortium</b>
<input type="checkbox"/>	European Commission
<input checked="" type="checkbox"/>	PUBLIC

**Status:**

<input type="checkbox"/>	For Information
<input type="checkbox"/>	Draft Version
<input type="checkbox"/>	Final Version (internal document)
<input type="checkbox"/>	<b>Submission for Approval (deliverable)</b>
<input checked="" type="checkbox"/>	Final Version (deliverable, approved on...)

<b>1. Introduction .....</b>	<b>3</b>
<b>2. Hardware design .....</b>	<b>4</b>
2.1 Installation.....	5
2.2 Antenna Installation .....	5
2.3 Power Supply and Loads Installation.....	5
2.4 Integrated Web Server .....	7
2.5 System Info .....	9
2.6 Processes .....	10
2.7 File Browser .....	11
2.8 Registry Editor .....	13
2.9 System Configuration .....	14
1) SNTP Settings.....	14
2) Installing Agents .....	15
2.10 Power Line Measurements.....	16
1) Frequency.....	16
2) Voltage RMS .....	16
3) Current RMS .....	16
4) Sag events detection and announcement.....	17
5) Overvoltage events detection and announcement .....	18
6) OverCurrent events detection and announcement.....	19
<b>3. Applications Developed.....</b>	<b>20</b>

## ***1. Introduction***

This deliverable provides a description of the Load Controller developed by ANCO and NTUA. The load controller is an embedded controller with communication and control capabilities. Furthermore the system can host the software agents that were developed in the WP B. The report presents some of the capabilities of the controllers as well part of the developed software. The reader should note that this work is correlation with the work in WP B as well the WP F (Kythnos test site).

More specific the ICT part of the plug n' play functionalities which is related to the DF agent of the Agent platform was presented in the DB3 deliverable. Therefore in this part we include different part of the plug n' play capability. We present the algorithm that uses the measurements of the Load controller and detects the network topology as well to detect the characteristics of a battery system. Finally the Reinforcement Learning Algorithm was also described in detail in the DB3.

## ***2. Hardware design***

The ANCO Intelligent Load Controller Unit (AN-ILC-01) is a system that can be used to monitor the status of a power line and take Voltage, Current and Frequency measurements. In addition it can remotely control up to 256 PLC A10 devices connected to the power line.

The AN-ILC-01 unit features a Wi-Fi interface that enables it to wirelessly connect to a Local Area Network. This eliminates the need of a data-cabling infrastructure and simplifies the installation of the units.

The core of the unit is an integrated computer module that runs the Windows CE 5.0 operating system. The integrated computer module is driven by the powerful Intel<sup>®</sup> Xscale<sup>™</sup> PXA255 processor at 400MHz and features 64MB of RAM and 32MB FLASH. Thus it is suitable for demanding applications.

A Java Virtual Machine that supports the Connected Device Configuration (CDC) can be installed in the Windows CE operating system. Therefore it is possible to execute the JADE-LEAP runtime environment on the AN-ILC-01. ANCO has developed a Java library that provides access to the power line control and measurements features of the unit. This library, together with the personal Java flavor of the JADE-LEAP, allows the user to launch power line agents.

ANCO provides editions of its Power Line Java library for both the CreMe JVM from NSICOM and the J9 JVM from IBM. Sample code that demonstrates the use of the library in the JADE-LEAP agent environment is also provided.

*2.1 Installation*

Each Intelligent Load Controller evaluation system consists of the following devices:

- One Intelligent Load Controller device AN-ILC-01 (ILC)
- One Stub type antenna
- Two A10 Switches

The ILC device is enclosed in a plastic box with two cables and an antenna connector permanently installed on it.

A LED indicator in the front panel lights when proper power is applied to the device. No ON/OFF switch is provided, thus the device is active as soon as power is applied.

*2.2 Antenna Installation*

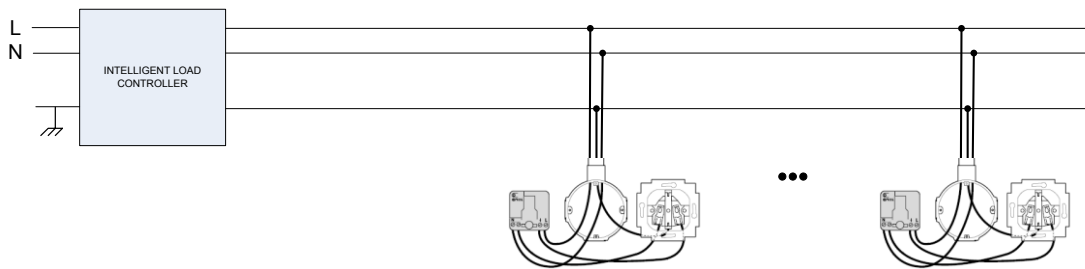
The user should install the accompanied antenna to the SMA connector for the Wi-Fi transceiver to operate. The ILC should be positioned in such a way that no obstacles are near the antenna for maximum performance.

*2.3 Power Supply and Loads Installation*

The ILC should be connected to the 230Vrms power utility grid via the wire with the power jack connector (ILC input).

The input voltage may range from 85Vrms to 264Vrm 50/60Hz.

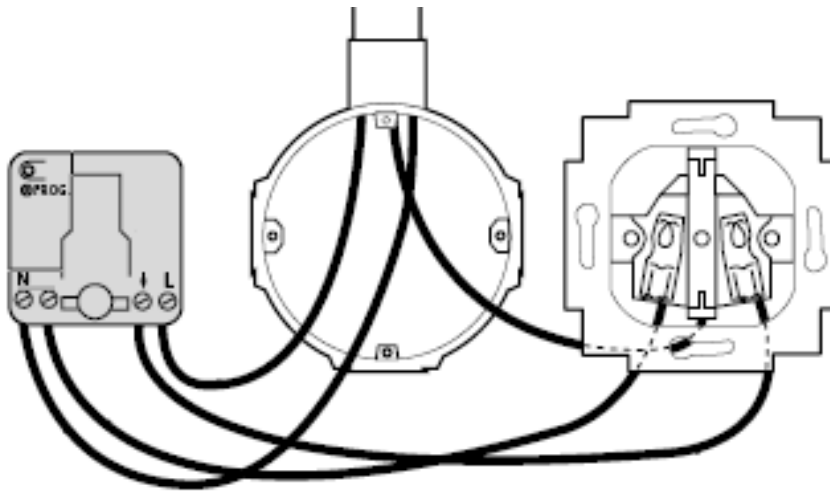
The loads of the ILC should be connected to the cable with the power socket connector (ILC output).



**Figure 1: Intelligent Load Controller Installation**

The current consumption of the loads should not exceed 16Arms in total.

In order to take advantage of the ON/OFF load control functionality of the ILC, each load should be connected to the output cable via one A10 switch according to Figure 2.



**Figure 2: A10 Switch Wiring Details**

Each load connected to an A10 switch should not exceed 16A.

**Please Note.**

- Each A10 switch attached to the output of the same ILC should have a unique ID.
- A10 switches belonging to different ILCs could share the same ID without the contingency of protocol conflicts.
- There is no overload protection inside the ILC for the Loads (e.g. thermal fuses, microswitches, e.t.c.). The user should provide with some means of protection externally.

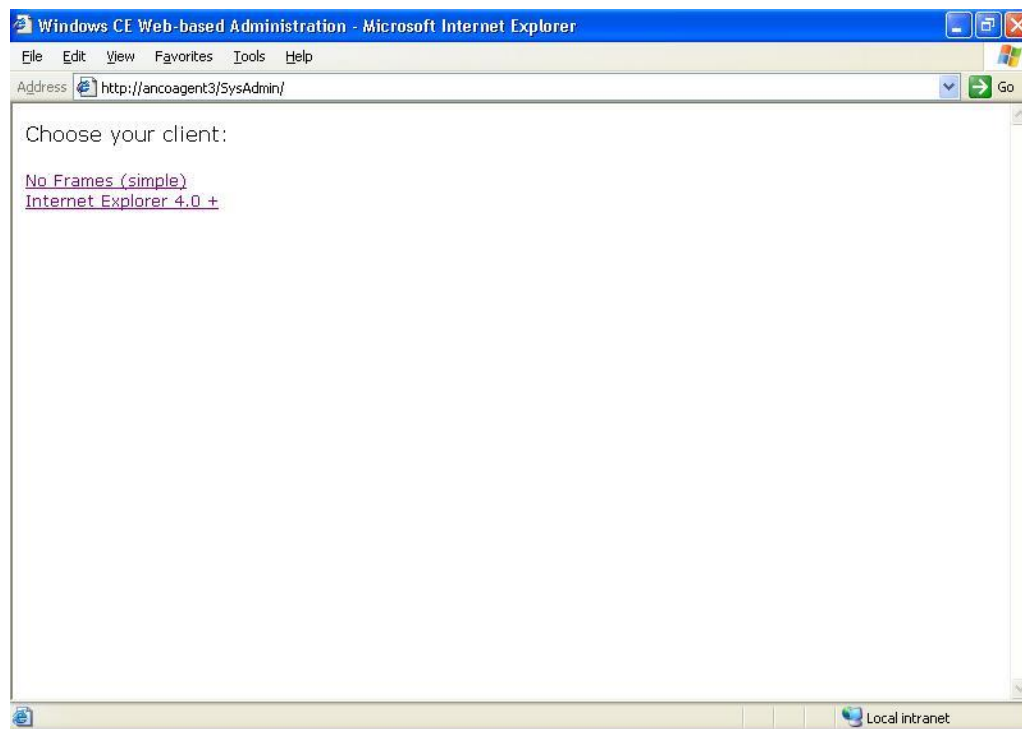
#### 2.4 Integrated Web Server

The Intelligent Load Controller can be accessed through its integrated Web Server. The unit must be attached to a Local Area Network with a running DNS and DHCP server. In addition a SNTP server is required so that all the units can share a common base time.

The Web Server of the unit with ID '1' is accessible at the address:

<http://AncoAgent1/SysAdmin>. The address for the units with different ID is altered accordingly. In addition to the Web Server, there is also the FTP Server, accessible at <ftp://AncoAgent1/>.

The Web Server can operate correctly only when the Internet Explorer browser is used. The following figure depicts the initial page of the server.



**Figure 3: Web Server Initial Page**

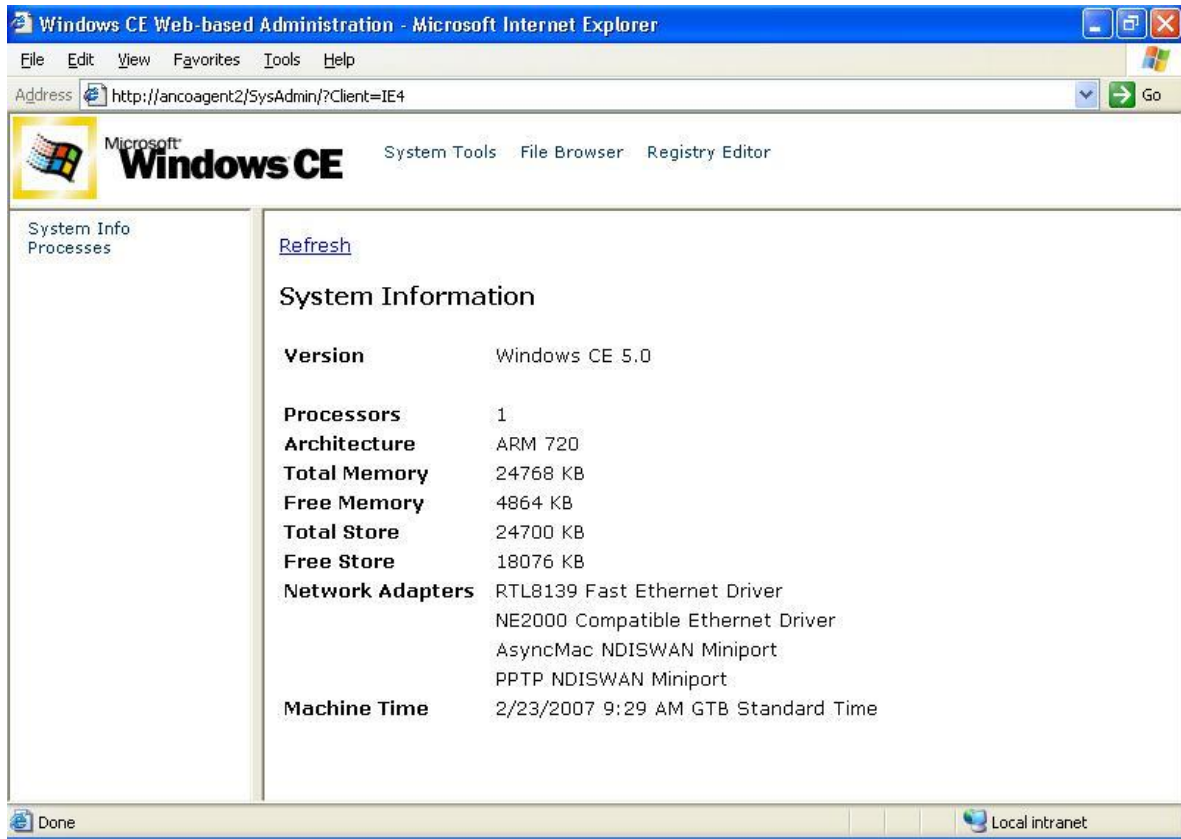
The user is asked to choose the type of its client browser. It is recommended to select the Internet Explorer 4.0+ option. When the selection is made, the Web Server presents the user with three different sets of tools:

- System Tools
- File Browser
- Registry Editor

Each of these sets of tools is selectable from the upper frame at the browser's window.

2.5 System Info

By clicking on the *System Tools* link at the upper frame, two links will be presented at the left frame: *System Info* and *Processes*.

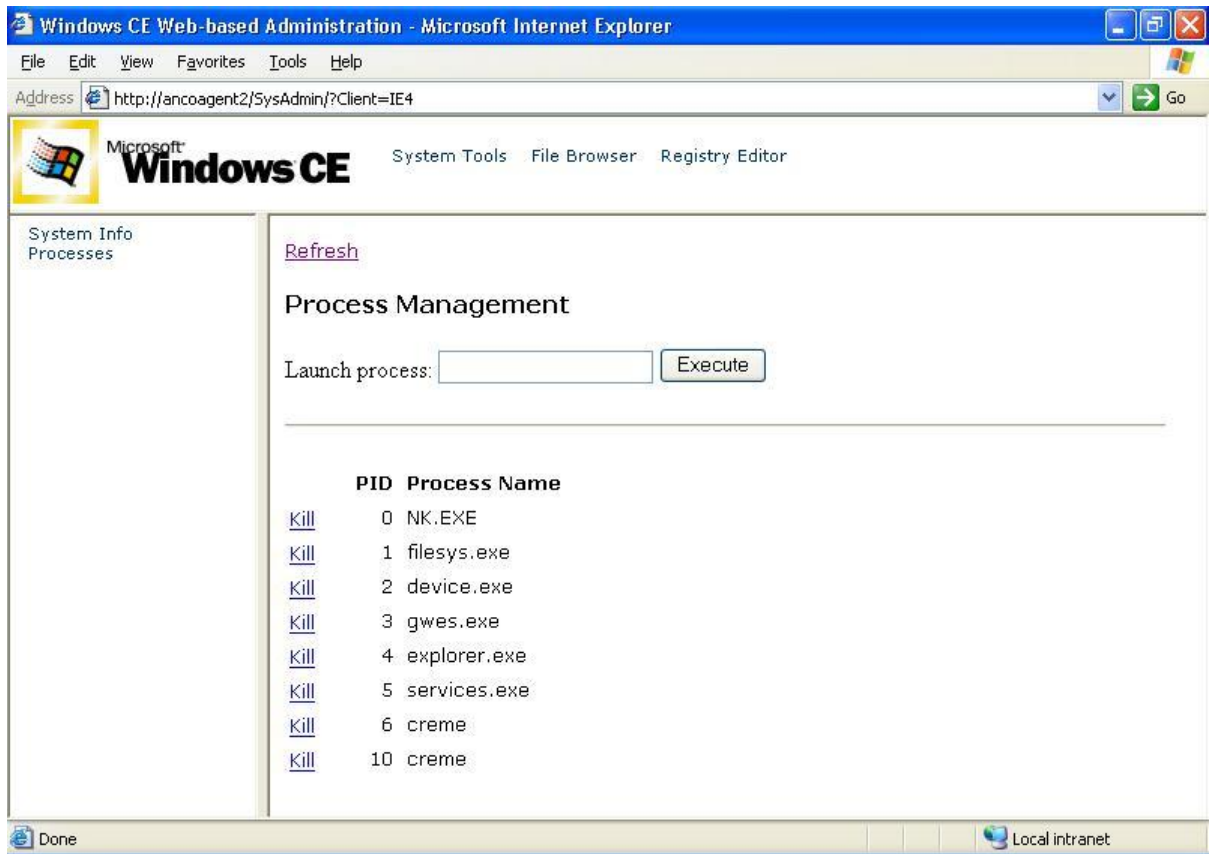


**Figure 4: System Info**

*System Info* link presents useful system information at the right frame of the browser’s window. The only field that is important for the user is the Machine Time, located at the bottom of the window. The user must verify that the unit is actually synchronized with the network.

## 2.6 Processes

The *Processes* link presents a list of the active processes in the system. The user can stop any of them by clicking on the Kill link next to them. It is recommended however not to make use of this feature.



**Figure 5: Processes**

The user can also start any application from this window. The name and the arguments of the application must be typed in the “*Launch process*” text box and then the “*Execute*” button must be pressed. The total length of the application’s name and its arguments must not exceed the 120 characters. To overcome this restriction, the AgentStart application can be used. The use of this application will be explained in a later section.

2.7 File Browser

The *File Browser* link provides access to the file system. The user can browse through the system’s folder, delete files and folders and upload new files. The File Browser should be used with extreme care and only for performing the operations described in the System Configuration section.

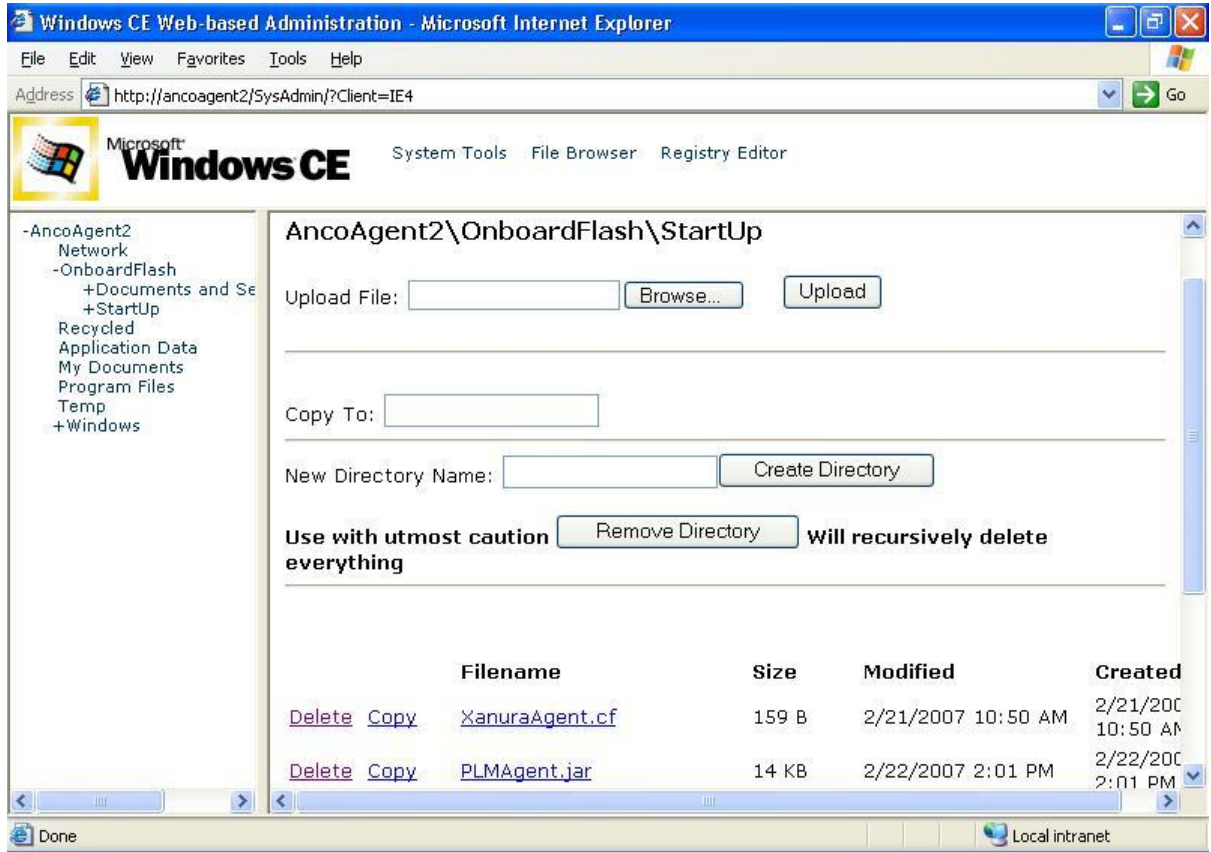


Figure 6: File Browser

The user should only modify the elements under the \OnboardFlash\StartUp folder. All other folders and files should be left as they are. Modifying any of it could make the system not operational.

The \OnboardFlash\StartUp folder is a placeholder for the JADE-LEAP agents. The contents of this folder are copied at start up to the root (\) directory. The root directory resides in RAM and copying the agent JAR files to it, guarantees maximum execution speed.

The \OnboardFlash\StartUp folder contains also the JadeLeap.jar file. This file has a size of approximately 1MB. The File Browser cannot be used to upload files of this size. Instead the FTP server must be used.

2.8 Registry Editor

The *Registry Editor* link provides access to the registry of the system. It should be used with extreme care and only for performing the operations described in the System Configuration section.

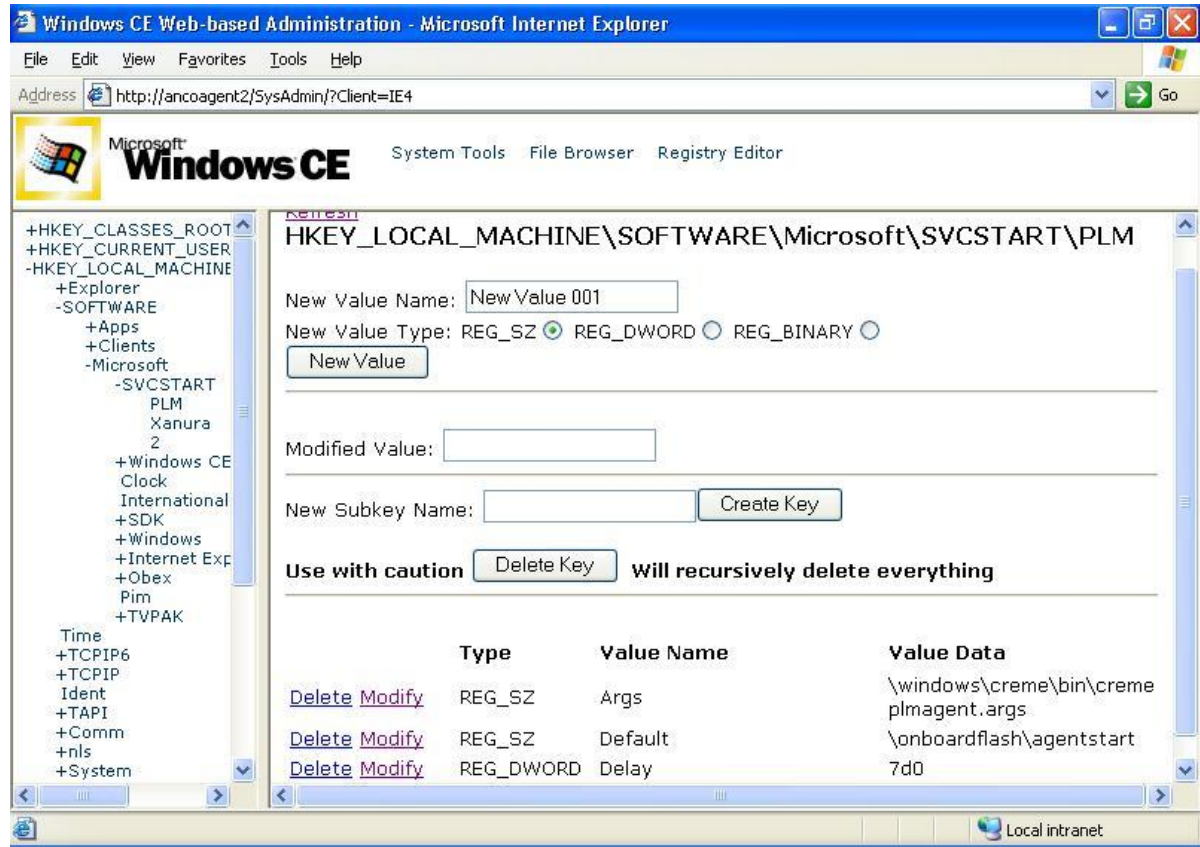


Figure 7: Registry Editor

2.9 System Configuration

1) SNTP Settings

The Intelligent Load Controller can connect to a remote SNTP server in order to synchronize their clocks. The address of the SNTP server is configured through the system’s registry.

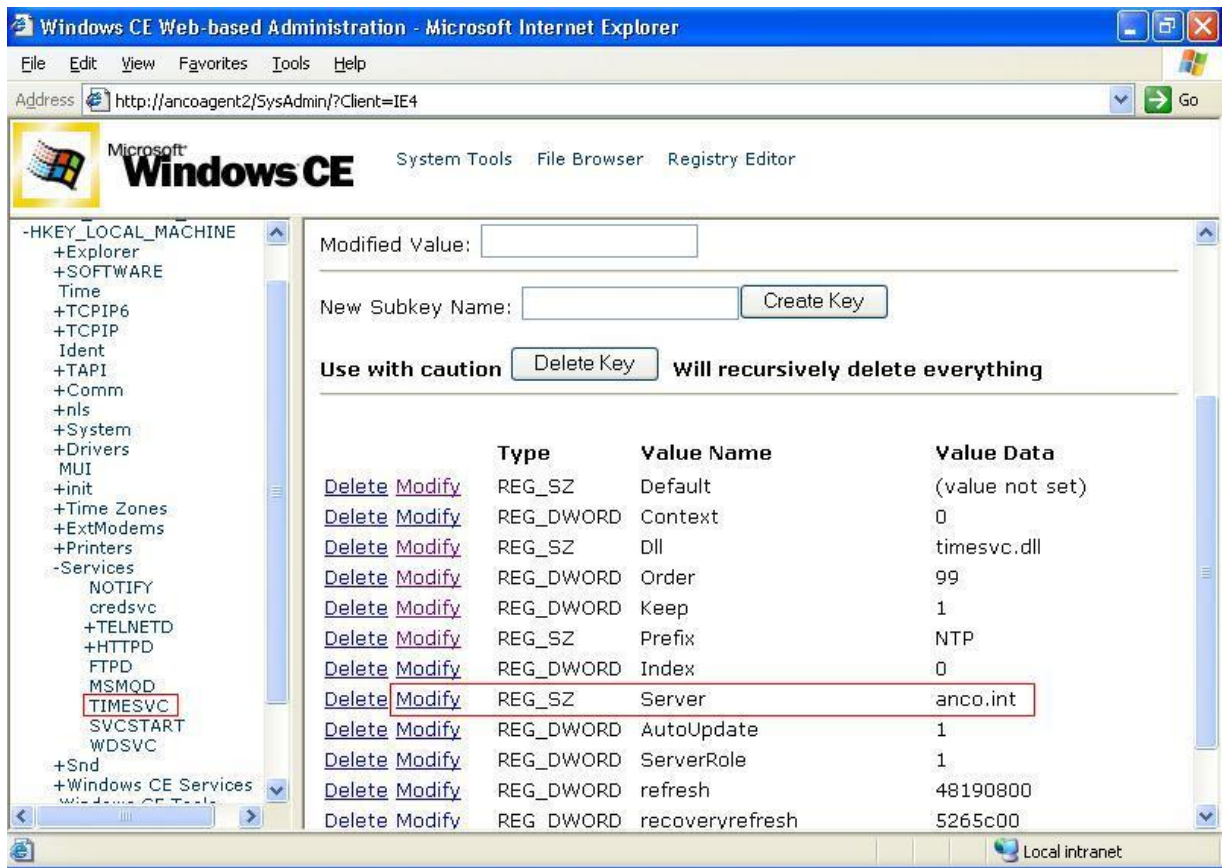


Figure 8: SNTP Configuration

The user must navigate to the key: *HKEY\_LOCAL\_MACHINE\Services\TIMESVC* and modify the *Server* value. This is achieved by writing the address of the SNTP server in the “Modified Value” text box and then clicking on the “Modify” link that corresponds to the *Server* value.

2) *Installing Agents*

In order to configure the Intelligent Load Controller to launch an agent at start-up the user must do the following:

- Place the JAR file of the agent at the \OnboardFlash\StartUp folder
- Create a file with the start up arguments for the agent (named e.g. agent.args) and place it to the \OnboardFlash\StartUp folder
- Configure the registry

The user must create a new sub-key under the

*HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\SVCSTART* key. The name of the sub-key (e.g. agent1) is not important; however there shouldn't be two sub-keys with the same name. Under this new sub-key the following values must be added:

*HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\SVCSTART\agent1*

Default	REG_SZ	\OnboardFlash\AgentStart.exe
Args	REG_SZ	\windows\creme\bin\creme agent.args
Delay	DWORD	7D0

AgentStart.exe is an application that takes as arguments the name of an executable and the name of a file, which contains in a single line the arguments list. AgentStart.exe will launch the executable with the supplied argument list. The executable to be called (i.e. the JAVA Virtual Machine) and the arguments list file can reside in RAM. AgentStart.exe will wait until they are copied to RAM and then will launch the process.

An example of the required registry settings is depicted at Figure 7.

### 2.10 Power Line Measurements

The Intelligent Load Controller built-in software provides the following functionalities regarding the power line measurements:

- Frequency measurement.
- Voltage RMS measurement.
- Current RMS measurement.
- Sag events detection and announcement.
- Overvoltage events detection and announcement.
- Overcurrent events detection and announcement.

#### 1) Frequency

The Intelligent Load Controller Java Agent application can read the instantaneous frequency at the desirable rate. The value returned is expressed in Hz. The measurement resolution is 2.2  $\mu$ s, which is about 0.013% at 60Hz.

#### 2) Voltage RMS

The Intelligent Load Controller Java Agent application can read the Voltage RMS value at the desirable rate. The value returned is expressed in Volts.

The Voltage RMS is calculated by sampling the power line at a rate of 3.5kSPS. The calculation involves squaring the signal, taking the average and obtaining the square root:

$$VRMS = V_{rms} = \sqrt{\frac{1}{N} \times \sum_{i=1}^N V^2(i)}$$

In order to minimize noise, the Vrms calculation is synchronized on the zero crossings of the voltage waveform. Moreover, the built-in software can return a rolling average calculated value, with programmable depth provided to MEAN\_SAMPLE\_No. Valid values range is from 1 to 100 and represent the number of the zero crossing measurements involved in the rolling average calculation.

#### 3) Current RMS

The Intelligent Load Controller Java Agent application can read the Current RMS value at the desirable rate. The value is expressed in Amperes.

The Current RMS is calculated by sampling the power line at a rate of 3.5kSPS. The calculation involves squaring the signal, taking the average and obtaining the square root.

In order to minimize noise, the Irms calculation is synchronized with the zero crossings of the Voltage waveform. Moreover, the built-in software can return a rolling average calculated value, with programmable depth provided to MEAN\_SAMPLE\_No. Valid values range is from 1 to 100 and represent the number of the zero crossing measurements involved in the rolling average calculation.

4) *Sag events detection and announcement*

The Intelligent Load Controller built-in software provides with the functionality of the power line Voltage Sag detection and announcement. Voltage Sag is defined as the condition where the absolute value of the line voltage drops below a certain peak value for a specified number of line cycles.

Should the Sag detection function is enabled the following parameters must be specified:

- *Vnom* : It is the voltage by which the system is designated to operate.
- *Voltage Reduction*: It is defined as the Vnom reduction factor under which it is considered as Voltage sag. The valid values range is from 0 to 1. The default value is 0.9 that means if Vrms drops below the  $0.9 \cdot V_{nom}$  a Sag event is announced.
- *SagCycle* : Specifies the number of consecutive line cycles where the Voltage signal peak should be below  $0.9 \cdot V_{nom}$ , in order to be considered as Sag event.

The software announces the Sag event start and termination. When a Sag event expires, the following parameters are available to the application:

- *SagDuration* : The duration of Sag event in ms.
- *SagDepth*: Specifies the difference between the minimum rms Voltage during the Voltage Sag and the nominal Voltage Vnom.

5) *Overvoltage events detection and announcement*

The Intelligent Load Controller built-in software provides with the option of the power line overvoltages detecting and announcement. An Overvoltage is defined as the condition where the absolute value of the line voltage exceeds a specified peak value.

Should the Overvoltage detection function is enabled the following parameters must be specified:

- *V<sub>peak\_level</sub>* : Specifies the Voltage peak threshold (in V) which if the line Voltage exceeds, is considered as Overvoltage.

The software announces the Overvoltage event start and termination. When Overvoltage event expires, the following parameters are available to the application:

- *OvervoltageDuration* : The duration of Overvoltage event in ms (Resolution: 10ms).
- *Voltage\_Peak*: Specifies the maximum absolute Voltage value observed during the Overvoltage event.

6) *OverCurrent events detection and announcement*

The Intelligent Load Controller built-in software provides with the option of power line overcurrents detection and announcement. An Overcurrent is defined as the condition where the absolute value of the line current exceeds a specified peak value.

Should the Overcurrent detection function is enabled the following parameters must be specified:

- *Ipeak\_level* : Specifies the Current peak threshold (in A), which if the line current exceeds, is considered as Overcurrent.

The software announces the Overcurrent event start and termination. When Overcurrent event expires, the following parameters are available to the application:

- *OvercurrentDuration* : The duration of Overcurrent event in ms (Resolution: 10ms).
- *Current\_Peak*: Specifies the maximum absolute line current value observed during the Overcurrent event.

### 3. Applications Developed

This functionality includes the operation where the agent that controls the loads, monitors them and tries to understand their behavior. The traditional load forecast is insufficient for loads in Low Voltage since the variations are highly stochastic. The approach presented here does not try to make a forecast of the consumption for the next hour but to define the behavior of the load in the next hours. For example it tries to define whether at 8:00 AM to 9:00 AM the load is steady, random or a combination of this two types.

Shortly the algorithm creates a population of fuzzy rules for each state of the system and after every operation an evolutionary algorithm returns a reward to the rule that suits best. The algorithm presented here uses fuzzy expressions for all possible behaviors of the load and tries to map the best rule and it uses real data to define the best rule and not an evolutionary algorithm.

For this operation there are certain assumptions:

- Each Microgrid has at least one group of controllable loads that are controlled by separate agents. For example each group can be a house and the various loads in a specific group are the loads of the house (lights, water heater etc).
- The second assumption is that we have partial knowledge of the environment meaning that we do not have power measurement on each load. However we do have measurement of the total consumption.

The second part is to create a canonical fuzzy description of the variable that we are interested to, for our case this is the overall consumption. This variable is described by three fuzzy variables. These variables are shown in figure 2. The fuzzy representation of all the variables is shown in figure 3 and expresses three states: the state near zero (S-function) which means the load definitely is not of the specific type (for example it is not Steady Load), the state near the middle (triangle function) which means that it cannot be validated what the type of the load and the state near one which means that the load is of the specific type.

The algorithm samples the load consumption every 5 minutes and checks how many times the measured load reaches the maximum power (*RateMaxPower*) and how many times reaches the minimum power (*RateMinPower*). These rates are also expressed by fuzzy variables with a form like the figure 3.

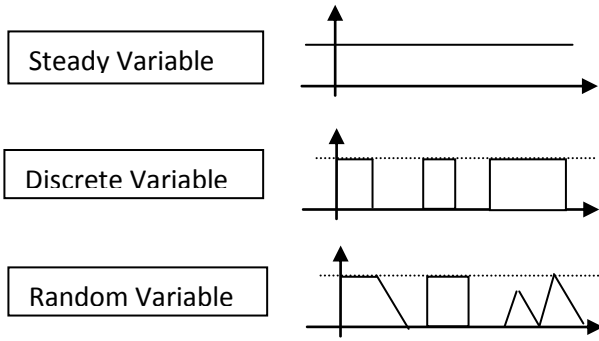


Fig 2. Fuzzy Variables.

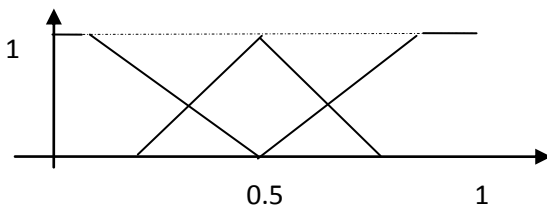


Fig 3. Fuzzy Representation of a Variable.

For each one of the three variables of figure 2 nine IF-THEN rules have been created like:

**IF *RateMaxPower* IS VERY HIGH AND *RateMinPower* IS VERY LOW THEN *STEADY* IS HIGH.**

The algorithm checks the total consumption of the load in the specific time step for the previous 50 steps. However it is significant to have records for all the possible states of the load otherwise the algorithm may reach wrong conclusion. After the collection of the data the algorithm evaluates which rule applies most to each variable. In a more advanced application the algorithm may create different tables for ordinary days and different tables for weekends or holidays. The development of the algorithm was based on the Java based toolbox [JFuzzy] which is free in the internet.

	Variable	Identification
1	Steady	100%
2	Discrete	97%
3	Random	90%

Table 3. Control levels of the MAS behaviours.

Table 3 presents the results of the algorithm of a software simulation. In every case when the load is Steady the identification is absolutely correct. There are some problems only with random type loads that in some cases are considered as Discrete and vice versa.

#### *Battery Identification.*

One significant part of a battery management system is the estimation of the battery electrical behaviour in different conditions of charging,. For this type of calculations for the Battery we consider the Shepherd model with the equation:

$$E = E_0 - IR_0 - K_i / \text{SOC} \quad (1)$$

Where  $R_0$  is the unknown internal resistance of the battery,  $K_i$  is a parameter,  $E$  is the Voltage of the battery,  $E_0$  is the open collector voltage and SOC is the State of Charge. The system acquires measurements during the operation and tries to find the  $(R_0, K_i)$  parameters by interpolation. The algorithm actually receives at least 50 values for different SOC. After having the model of the battery the system is ready to estimate the effect in the SOC for the various values of the current.

This functionality will be part of a class of functionalities that will form an integrated battery management system. However it is beyond of the scope of this paper to present this type of functionalities. The presented formula is presented since it is necessary for the calculations of the upper level of our system.

*Network topology identification.*

One significant issue for the Microgrids is the knowledge of the network topology. The problem is complex since the demand for “plug n’ play” capabilities suggest that topology knowledge should not be inserted from the installer. Therefore the problem is whether the agents can identify by themselves the electrically neighboring agents.

The approach that will be presented here is based on the statistical analysis and more specifically on calculating the correlation coefficient for two random variables X and Y:

$$r = \frac{\text{Cov}(X,Y)}{\sqrt{V[X]V[Y]}} \quad (2)$$

The correlation coefficient provides an indication on whether we can estimate the variable X if we know the variable Y and vice versa. If  $r=0$  then we cannot make any estimation but if  $|r|=1$  then we can be positive about the value of the other variable.

When the correlation coefficient is unknown, as in most cases then it can be estimated based on pairs of observations:  $(x_i, y_i)$  ( $i=1, \dots, n$ ). If the pairs  $(x_i, y_i)$  are binomial normal set [διμεταβλητό κανονικό πληθυσμό] then the  $r$  is:

$$r = \frac{\sum_1^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_1^n (x_i - \bar{x})^2 \sum_1^n (y_i - \bar{y})^2}} \quad (3)$$

And is called linear correlation factor [δειγματικός γραμμικός συντελεστής συσχέτισης]. The value  $r = \pm 1$  are occurred only when pairs  $(x_i, y_i)$  ( $i=1, \dots, n$ ) are in the same line.

For testing the algorithm two case networks were created as shown in figure 4 in the Power system Laboratory of NTUA. For tests 2,5mm<sup>2</sup> conventional NYA cable was used. The voltage was 150V. The reason for this was that the cable was not long enough and by reducing the voltage we would have large voltage drop. The various loads are controlled by a PLC (programmable logical controller) that operated the switches randomly. The difference between the two cases is the position of Load E.

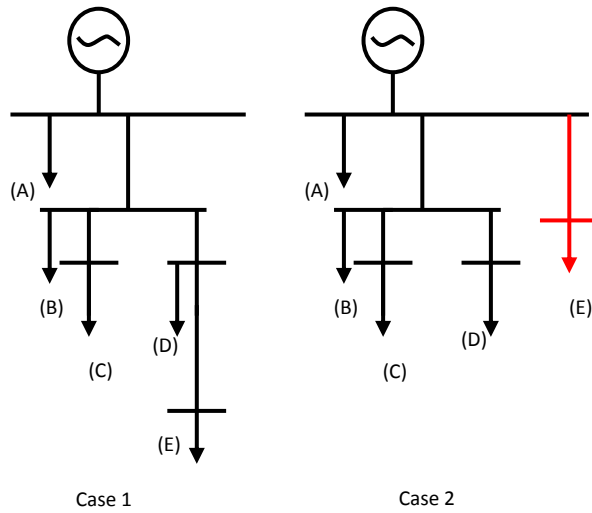


Figure 4

The sampling for the correlation coefficient was 0.5Hz and the vector has 100 values.

The results for the calculation on both cases are:

a) Case 1

<b>r</b>	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>
<b>A</b>	-	<b>0.95</b>	<b>0.95</b>	<b>0.95</b>	<b>0.94</b>
<b>B</b>	<b>0.95</b>	-	<b>1</b>	<b>1</b>	<b>1</b>
<b>C</b>	<b>0.95</b>	<b>1</b>	-	<b>1</b>	<b>1</b>
<b>D</b>	<b>0.95</b>	<b>1</b>	<b>1</b>	-	<b>1</b>
<b>E</b>	<b>0.94</b>	<b>1</b>	<b>1</b>	<b>1</b>	-

b) Case 2

<b>r</b>	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>
<b>A</b>	-	<b>0.95</b>	<b>0.95</b>	<b>0.95</b>	<b>0.96</b>
<b>B</b>	<b>0.95</b>	-	<b>1</b>	<b>1</b>	<b>0,8</b>
<b>C</b>	<b>0.95</b>	<b>1</b>	-	<b>1</b>	<b>0,8</b>
<b>D</b>	<b>0.95</b>	<b>1</b>	<b>1</b>	-	<b>0,8</b>
<b>E</b>	<b>0.96</b>	<b>0,8</b>	<b>0,8</b>	<b>0,8</b>	-

From the results it is obvious that the correlation coefficient is changed significantly for case 2. However in case 1 the results are not all 1 since cable have not negligible resistance making the behavior not linear.