DAEnetIP4

Web enabled IP Controller

User Manual
Date: 08 Aug 2022

Content

1. Features

DAEnetIP4 is the next generation multifunctional Ethernet device (IP controller) for management and control. It could be used for industrial and home automation, access control, fire and security systems and embedding in other systems. It is suitable also for controlling relay boards, monitoring different analogue sensors via internet.

DAEnetIP4 features include:

- 10/100 Mb Ethernet interface;
- Auto-MDIX;
- Extended supply voltage range (7.5V - 25V) DC
- Low power consumption - 40mA at 24V DC;
- 16 (2x8) Digital Outputs;
- 8 Analog Inputs (10 bit resolution);
- 8 Digital Inputs (16 bit counters);
- 2 PWM outputs (10 bit resolution);
- Real Time Clock (RTC) for schedule (calendar) stand-alone work;
- Extra status led;
- Web server with secure login authorization;
- SNMPv1 for configuration/monitoring (get,set,bulk,table,getnext,walk);
- Secure HTTP/XML/JSON API protocol support for read/write Inputs/Outputs;
- Supported protocols: ARP, IP, HTTP, ICMP (ping), DHCP, DNS;
- Access protection (by IP and MAC address);
- Option for saving Digital Outputs states and restoring on reset;
- Scaling (linearization) for the analog inputs to show values in suitable units;
- Single Input may be forwarded to control many Digital Outputs;
- Sending traps according to Analog Input level and Digital Inputs state;
- Working temperature range: 0ºC to 70ºC.

2. Application examples

Bellow are shown ideas how **DAEnetIP4** could be used. The examples are only conceptual and an additional equipment is required in actual implementations:

- Electrical appliances remote control applications

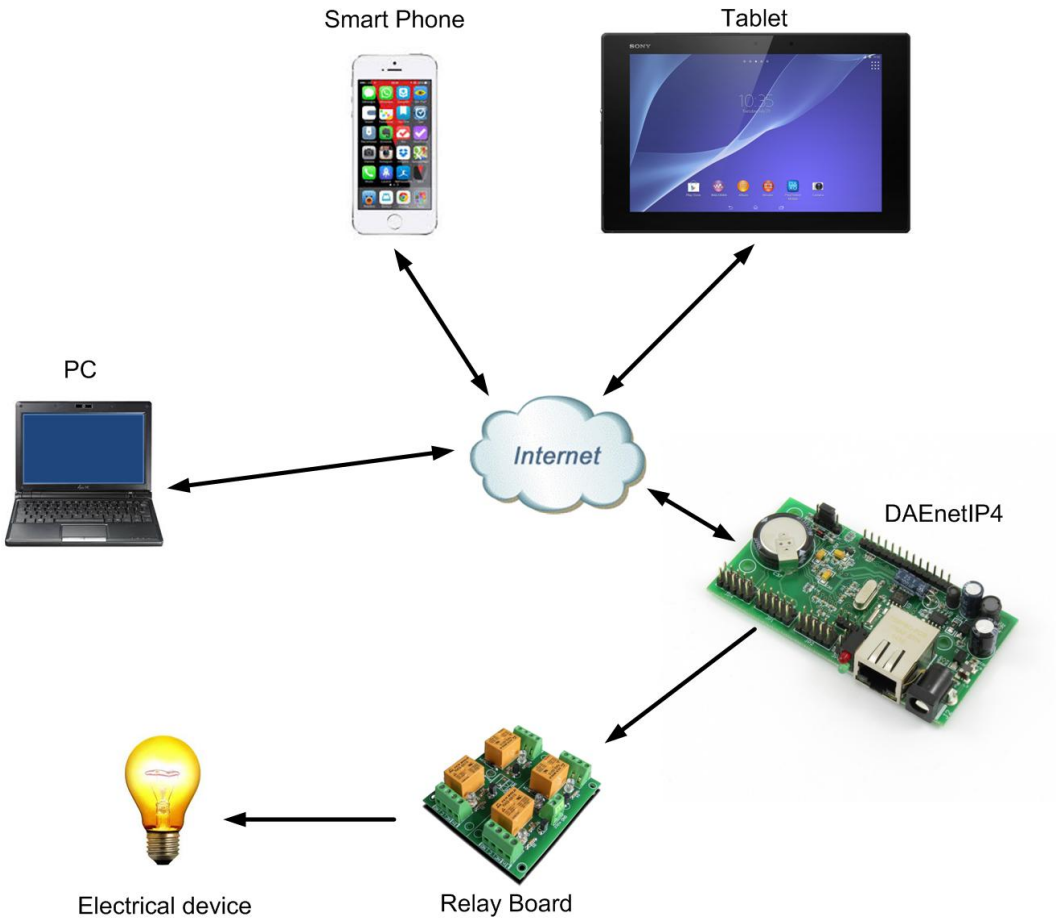

Figure 1. Controlling electrical appliances remotely

DAEnetIP4 can be used to control electrical appliances with combination of mechanical relays, solid state relays, contactors and so on. It has 16 Digital Outputs so it can control 16 electrical devices independently. Supports various integration protocols (SNMP, HTTP/XML, Web-browser access) so these devices can be accessed via any modern device from all over the world.

- Monitoring and logging applications

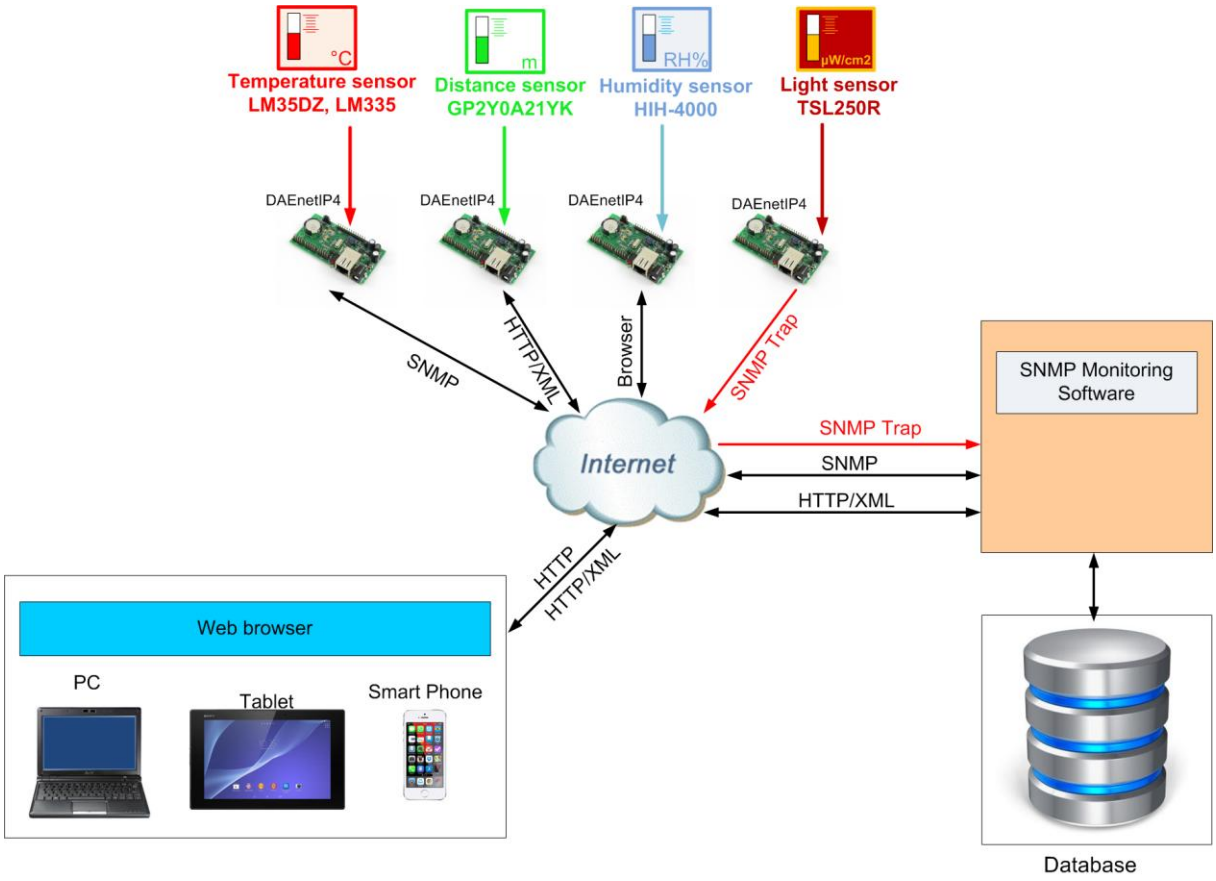

Figure 2. Monitoring sensors

DAEnetIP4 can be used with success into monitoring and logging systems. It has 8 x 10 bit Analog Inputs (10-bit resolution) which can be level-extended in order to monitor temperature, humidity, distance, light and so on.

- Standalone applications with sensors and electrical devices

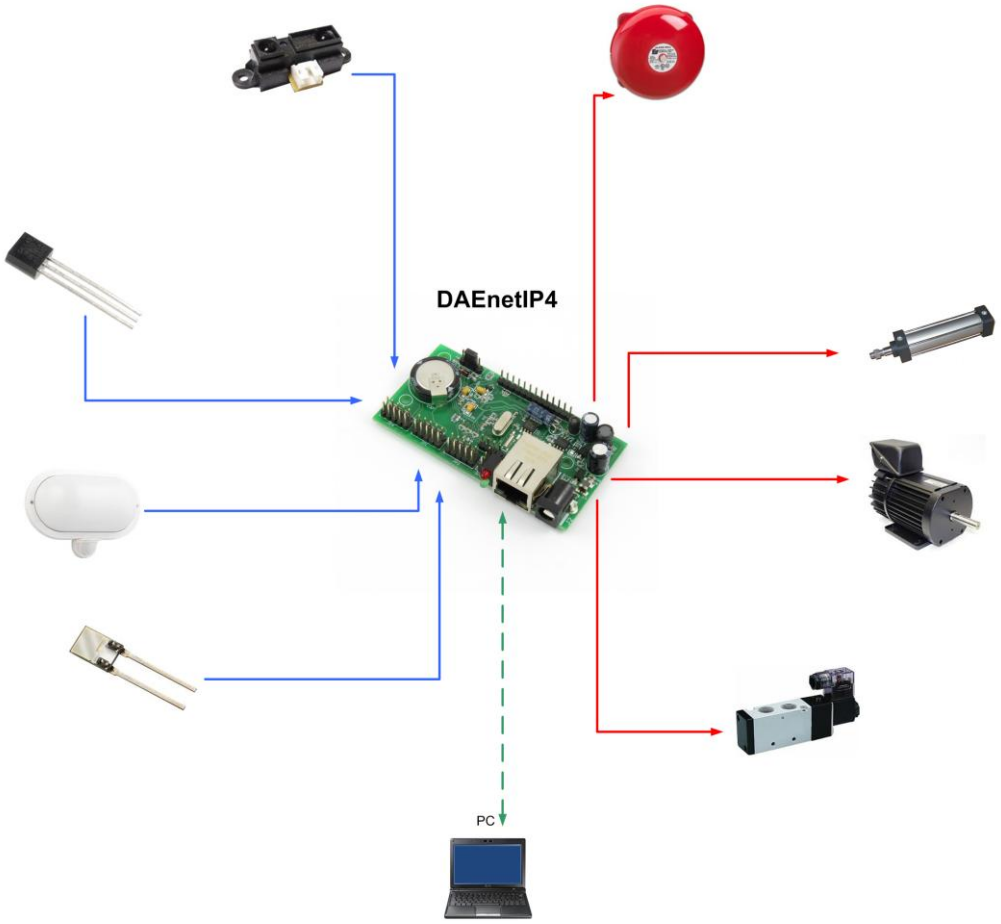

Figure 3. Controlling electrical devices depending on sensors values

The controller supports mode in which the Analog/Digital Inputs (sensors) can control the Digital Outputs (electrical devices).

- Counters in shops

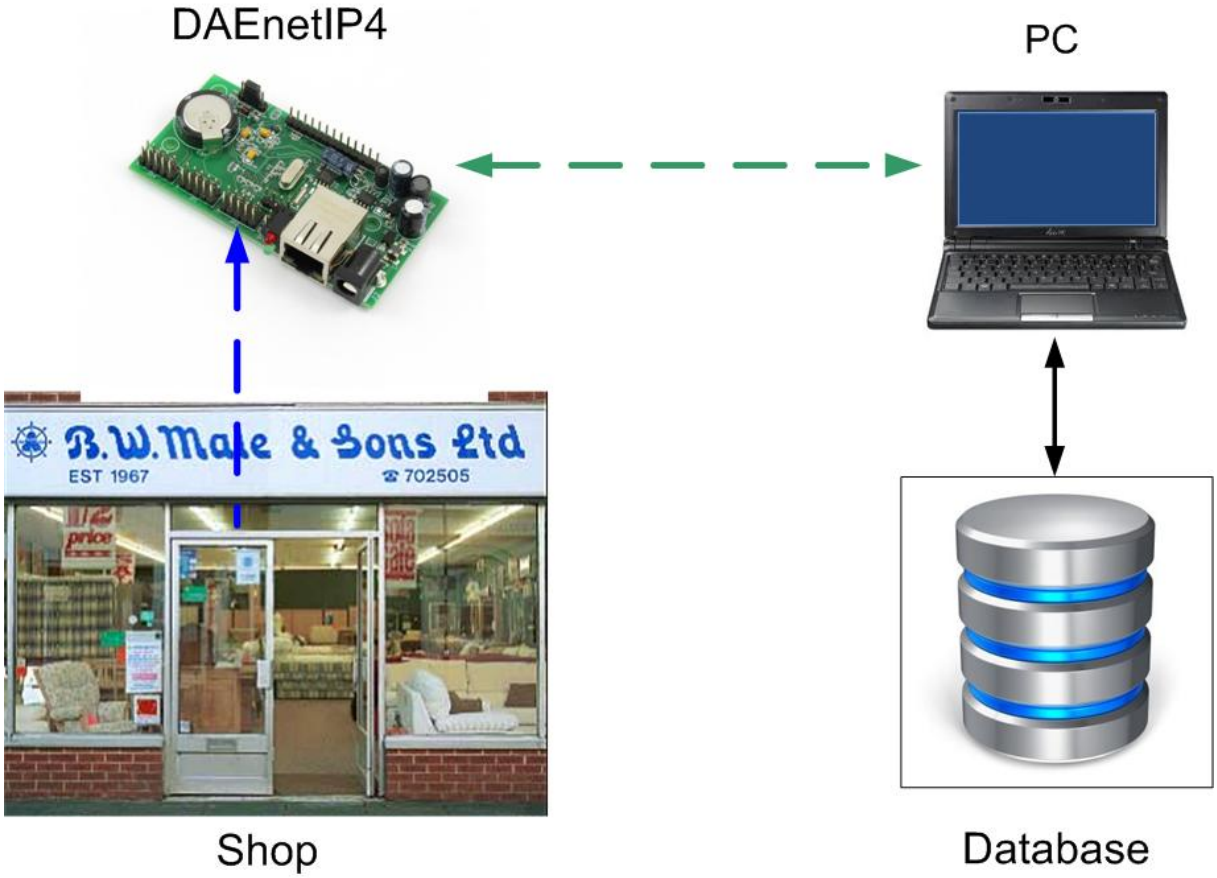

Figure 4. DAEnetIP4 used in application for counting customers in shops

- **DAEnetIP4** provides 8 x 16 bit counters (from 0 up to 65535) which can be used to count various events - for example detect when a person enters in a shop through the door. **DAEnetIP4** increment its counter and saves this in its RAM memory. With suitable software and database it could be easily organized a monitoring and statistic system without concerns that even an event will be missed.

- Web based thermo-regulator

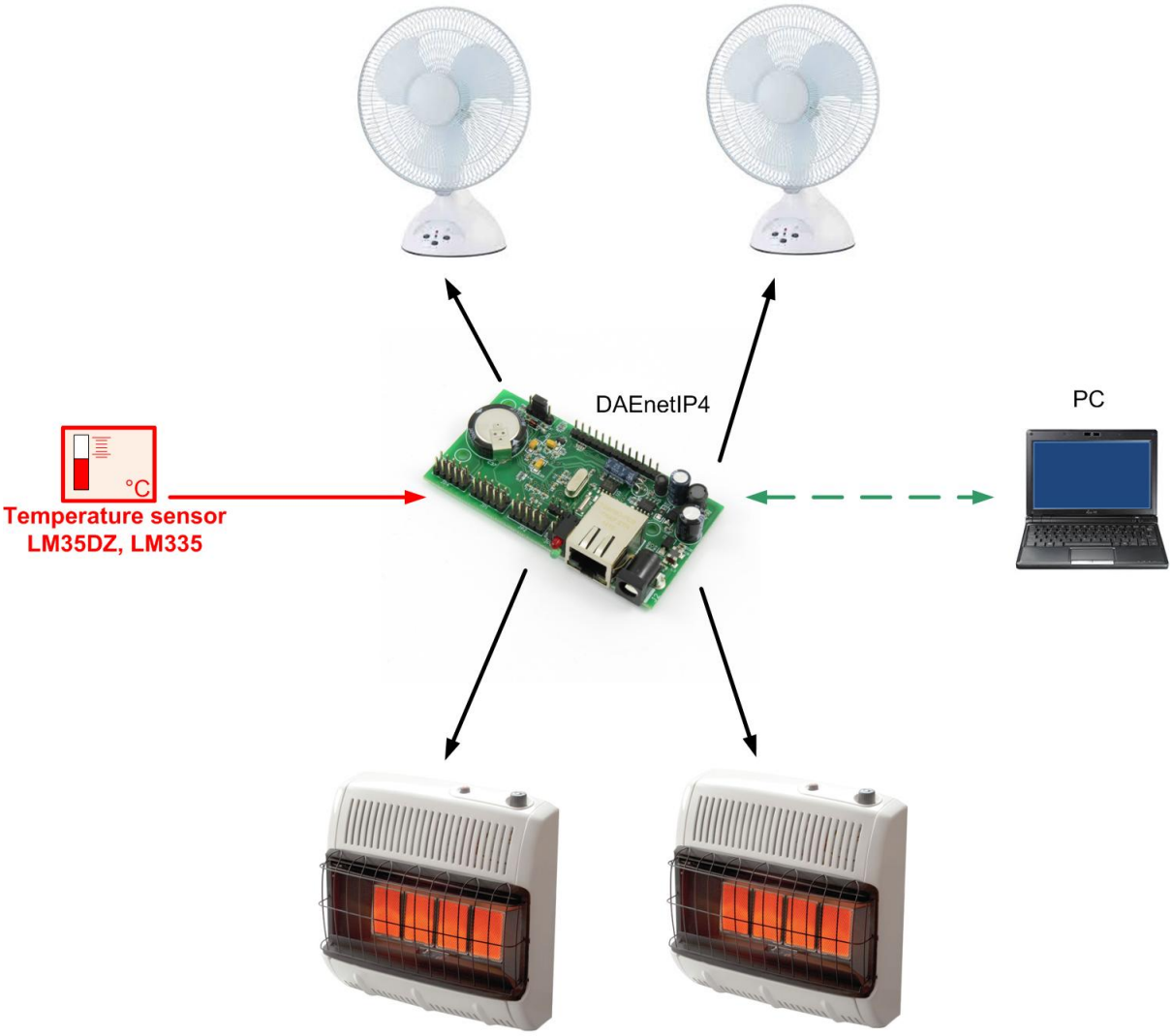

Figure 5. Web based thermo-regulator

Every **DAEnetIP4** Digital Output can be set to work in "Regulator" mode where it can be controlled just from an Analog Input. The controller can be adjusted to set different outputs upon the input level from single Analog Input (one input can be configured to control many outputs). For example, one output can be set for heating, another for cooling and etc. All the process may be monitored online and this makes **DAEnetIP4** suitable for standalone Web enabled thermo-regulator.

- Home automation

Figure 6. DAEnetIP4 can be used in home automation systems

DAEnetIP4 can be used for home automation applications as well.

- It has 2 x PWM (Pulse Width Modulation) outputs which can be used (with extension board) for lights dimming;
- Thanks to it's week schedule feature (it has built in RTC), it could turn on/off appliances up to 30 times per day;
- Setting with pulse. Every Digital Output can be configured to generate a single pulse (from 0 up to 65553 seconds) which could be suitable for controlling garage doors, windows, roofs and for any other application where a certain period of time should be maintained;
- **DAEnetIP4** supports mode of controlling the Digital Output from Digital Input (buttons, switches...), Analog Input (sensors for temperature, humidity, distance, light...), RTC (week schedule), Manual Control (browser, SNMP, HTTP/XML) at the same time. This means the output could be controlled for example from smart-phone, wall switch and light sensor at the same time and with priority is the last action.
- Using the "Regulator" mode for the Digital Outputs can ensure for example opening and closing the door blinds based on the level of the outside sunlight;

- Irrigation systems

Figure 7. DAEnetIP4 can be used in home irrigation systems

One of the possible applications for DAEnetIP2 is to be integrated in irrigation systems. It features two different modes:

- Start and stop irrigation based on particular time. As **DAEnetIP4** has its own RTC (real time clock) and build in back-up power (for days) it can turn on and off the system up to 30 times per a single day - it has organized a week schedule based table inside. The clock can be synchronized via internet from NTP server during a given period of a time.
- Start and stop irrigation based on humidity level. **DAEnetIP4** can also set the solenoid valves upon the level from humidity or rain sensor which ensures that if it is rainy it won't start the system. This is because the Analog Inputs of the controller can control the Digital Outputs independently without computer (standalone mode).

3. Technical parameters

Table 1. Technical parameters

Parameter	Value
Size (L / W / H), mm	85 / 48 / 17
Power supply voltage, V DC	7.5 - 25
Current consumption, mA	40mA at 24V, 70mA at 12V, 90mA at 9V
Digital outputs count	16 (2 x 8)
Digital outputs voltage, V	0 or 3.3
Analog Inputs count	8
Analog Inputs reference voltage, V	2.048
Analog Inputs resolution, bits	10
Digital Inputs count	8
Digital Inputs voltage range, V	0 up to 5.5
Counters count*	8
Counters resolution, bits	16
PWM outputs count	2
PWM frequency, KHz	40
PWM output voltage, V	0 up to 3.3
PWM resolution, bits	10
Default settings jumper	Yes
Reset jumper	Yes
Save I/O states and load on boot	Yes
DHCP	Yes
DNS	Yes
Hardware Real Time Clock (RTC)	Yes
Network parameters	IP/Mask/Default gateway
MAC lock (protection)	Yes
SNMPv1	Yes (set,get,table,walk,getnext,bulk)
Read-Write Community String	Yes
Read-Only Community String	Yes
SNMP traps	Yes
Secure HTTP/XML access	Yes
SNMP I/O access commands	Yes
Web server for configuration/access	Yes

* Combined with Digital Inputs

4. Connectors and ports (interfaces)

4.1. DAEnetIP4 ports

Bellow is shown a picture with the device connectors, ports and led indicators.

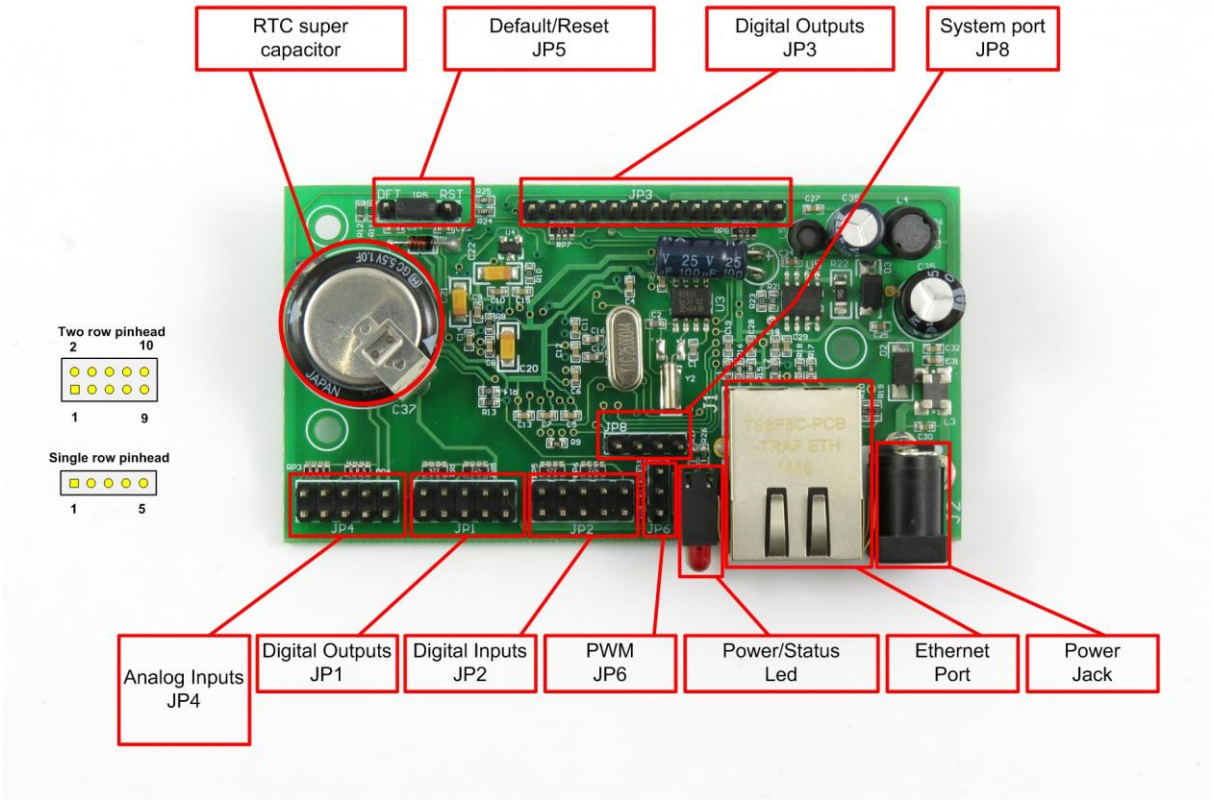

Figure 8. Device overview

4.2. DAEnetIP4 ports, led-s and jacks description

The first pin of every I/O port is marked with square footprint and all the rest pins are with circle footprints.

Table 2. Digital Outputs port JP1

Pin N	Bit	Function	Direction
1	0	General Purpose Output #1	OUT
2	1	General Purpose Output #2	OUT
3	2	General Purpose Output #3	OUT
4	3	General Purpose Output #4	OUT
5	4	General Purpose Output #5	OUT
6	5	General Purpose Output #6	OUT
7	6	General Purpose Output #7	OUT
8	7	General Purpose Output #8	OUT
9	-	GND	-
10	-	3V3	-

Table 3. Digital Inputs port JP2

Pin N	Bit	Function	Direction
1	0	General Purpose Input #1	IN
2	1	General Purpose Input #2	IN
3	2	General Purpose Input #3	IN
4	3	General Purpose Input #4	IN
5	4	General Purpose Input #5	IN
6	5	General Purpose Input #6	IN
7	6	General Purpose Input #6	IN
8	7	General Purpose Input #7	IN
9	-	GND	-
10	-	3V3	-

Table 4. Digital Outputs port JP3

Pin N	Bit	Function	Direction
1	8	General Purpose Output #9	OUT
2	9	General Purpose Output #10	OUT
3	10	General Purpose Output #11	OUT
4	11	General Purpose Output #12	OUT
5	12	General Purpose Output #13	OUT
6	-	+VCC	-
7	-	+VCC	-
8	-	GND	-
9	-	GND	-
10	-	GND	-
11	13	General Purpose Output #14	OUT
12	14	General Purpose Output #15	OUT
13	15	General Purpose Output #16	OUT

Table 5. Analog Inputs port JP4

Pin N	Bit	Function	Direction
1	-	ADC Channel #1	AIN
2	-	ADC Channel #2	AIN
3	-	ADC Channel #3	AIN
4	-	ADC Channel #4	AIN
5	-	ADC Channel #5	AIN
6	-	ADC Channel #6	AIN
7	-	ADC Channel #7	AIN
8	-	ADC Channel #8	AIN
9	-	GND	-
10	-	Vref (+2.048V)	-

Table 6. System port JP5

Pin N	Bit	Function	Direction
1	-	Default (factory) settings	IN
2	-	GND	-
3	-	GND	-
4	-	Reset	IN

Table 7. PWM port JP6

Pin N	Bit	Function	Direction
1	-	PWM1	OUT
2	-	PWM2	OUT
3	-	GND	-

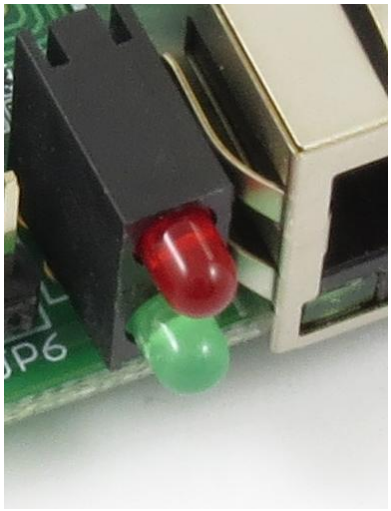

Figure 9. Power and status led

The top led next to the RJ-45 indicates power presence (usually red). The bottom led is an indicator for status (usually green).

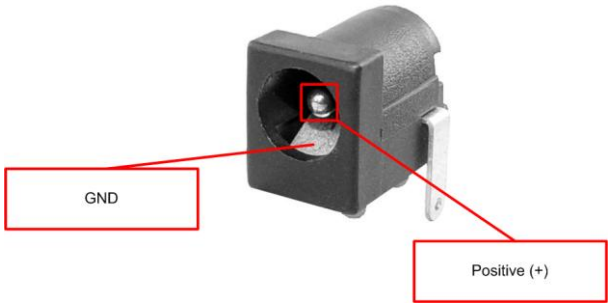

Figure 10. Power jack polarity

The inner pin of the power supply jack is the positive (+VCC) and the outer is the negative (GND).

5. Installation

5.1. Connect

- This device must be installed by qualified personnel;
- This device must not be installed directly outdoors;
- Installation consists of mounting the device, connecting to an IP network, connecting the I/O, providing power and configuring via a web browser.

5.2. Power supply requirements

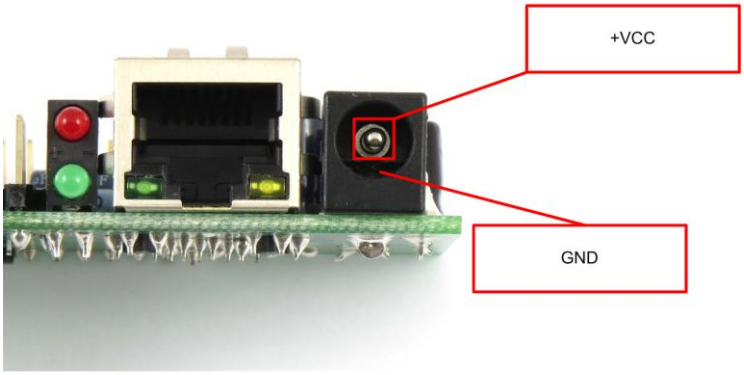

Figure 11. Location of **DAEnetIP4** power jack

DAEnetIP4 has the following current consumption:

- 40mA at 24V DC
- 70mA at 12V DC
- 90mA at 9V DC

It is recommended the supply source for **DAEnetIP4** to be with the following parameters:

- Supply voltage: 7.5V - 25V DC;
- Current: minimum 100mA;
- It must be stabilized and filtered;
- Type: center positive (the inner pin of the power supply adaptor jack must be +VCC).

Figure 12. How the power supply cable must looks like

Additionally, you can check if the supply adaptor has this sign:

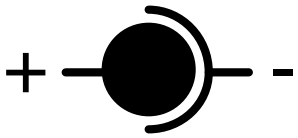

Figure 13. The power supply must be marked with this sign

- **DAEnetIP4** has protection against reverse polarity which is actually diode in parallel of the supply jack but it is still **not recommended** to reverse the voltage polarity!
- **DAEnetIP4** does not accept AC supply voltage. It is highly recommended to check the power supply source parameters before turning on the module.
- The power supply equipment shall be resistant to short circuit and overload in secondary circuit.
- When in use, do not place the equipment so that it is difficult to disconnect the device from the power supply.

5.3. Network connection

DAEnetIP4 supports AUTO-MDIX so either "crossover" or "straight-through" network cable can be used.

Figure 14. UTP Cable

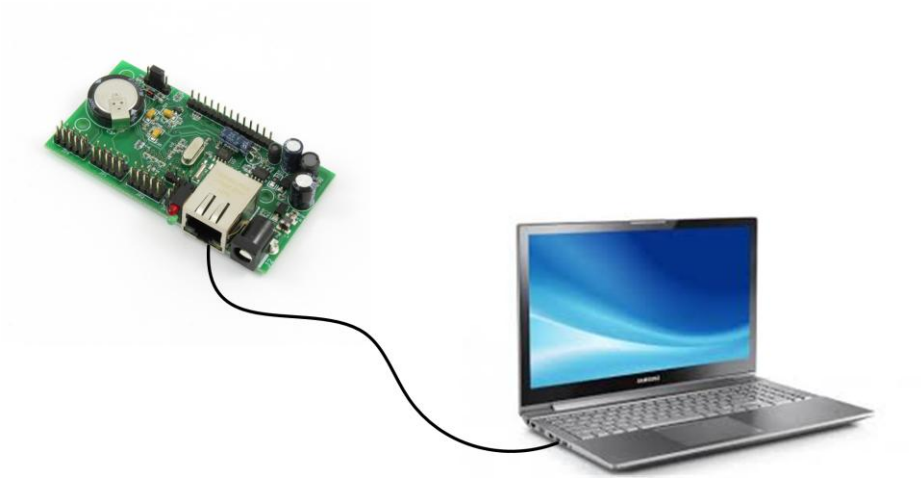

Figure 15. Connecting **DAEnetIP4** to a computer directly (recommended initial connection)

Figure 16. Connecting **DAEnetIP4** to a wireless router.

5.4. Communication setup

DAEnetIP4 is shipped with the following default parameters:

- IP address: **192.168.1.100**
- Subnet mask: **255.255.255.0**
- Gateway: **192.168.1.1**
- Web password: **admin**

Initially it is recommended to connect the module directly to the computer.

Next you have to change your PC's IP address.

You can Google how to change you computer IP settings or just visit this web page: http://www.howtochangeipaddress.com/changeip.php

For Windows 7 OS for example you can do that in the following way:
Navigate to *Control Panel -> Network and Internet -> View network and status tasks -> Change adapter settings*

Then just select the local area connection with right click and select *Properties*:

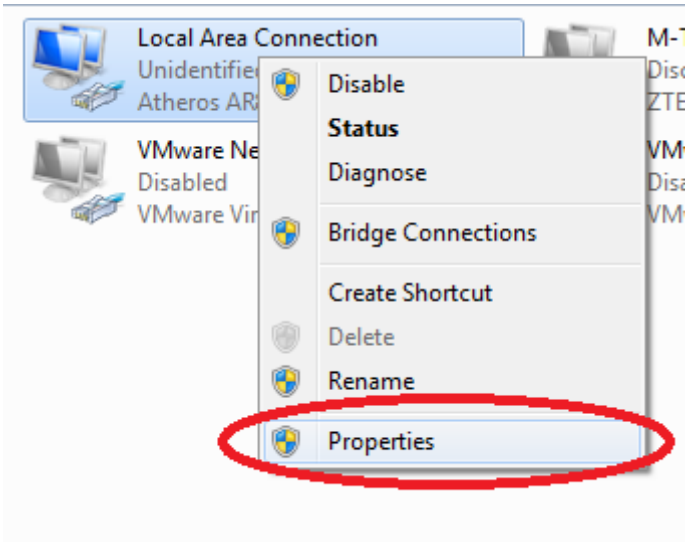

Figure 17. LAN card properties

The next step is to enter into IPv4 properties.

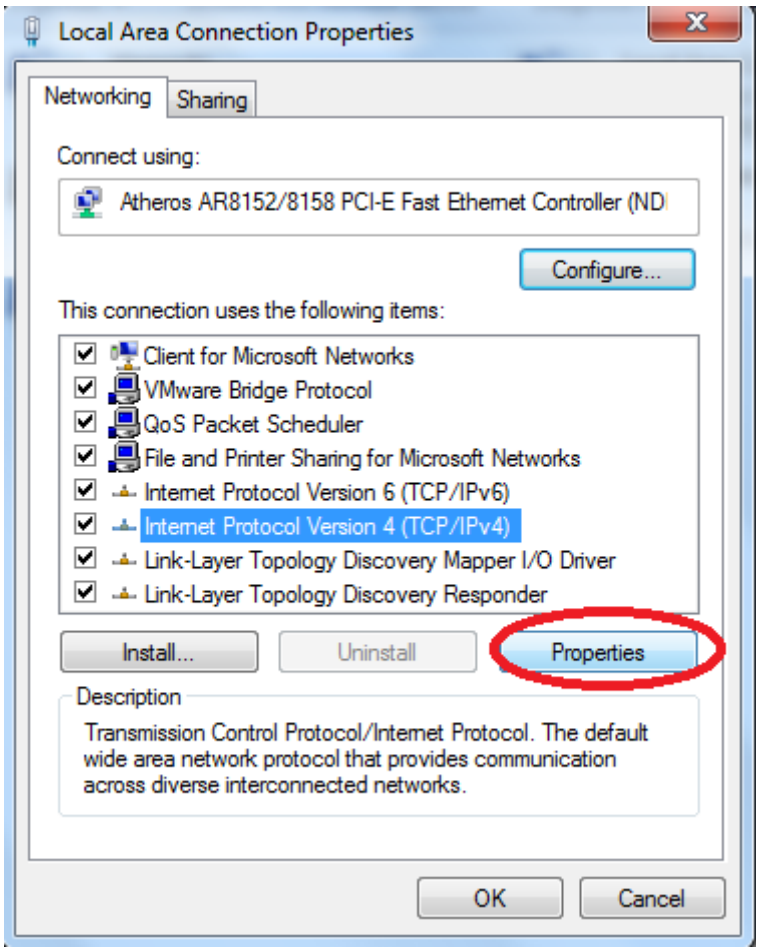

Figure 18. Enter in IPv4 properties section

Set the IP address of your PC to be in the same network.

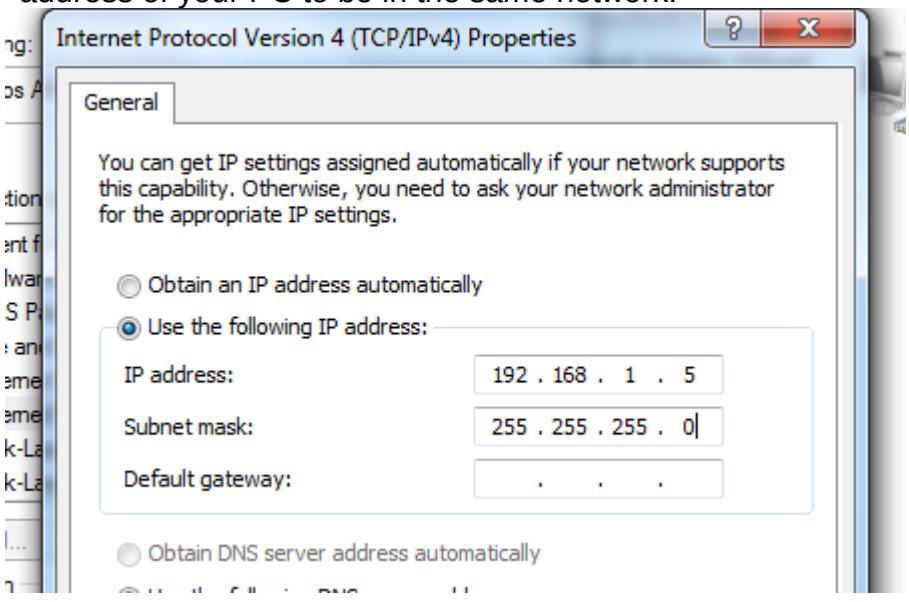

Figure 19. Set the IP address

Finally, in order to access **DAEnetIP4** just type in your browser 192.168.1.100

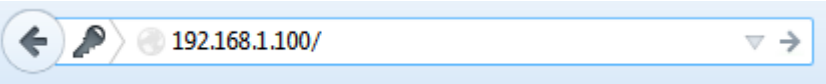

Figure 20. Open the device via browser

If the network settings are O'K, the log-in page should appear:

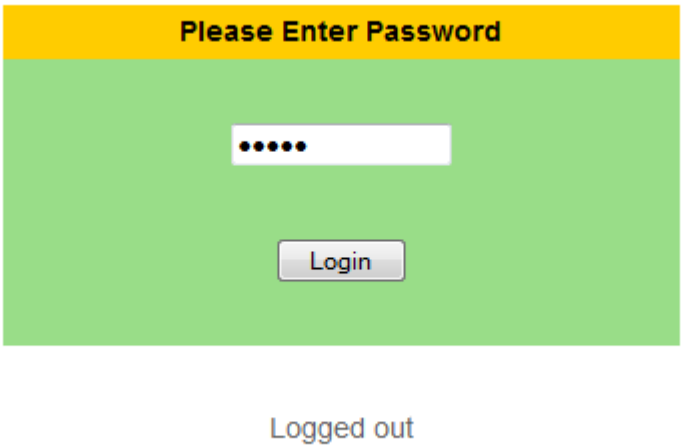

Figure 21. Login page

DAEnetIP4 modules connected locally can be easily scanned and found via the tool *Denkovi Finder* as well.

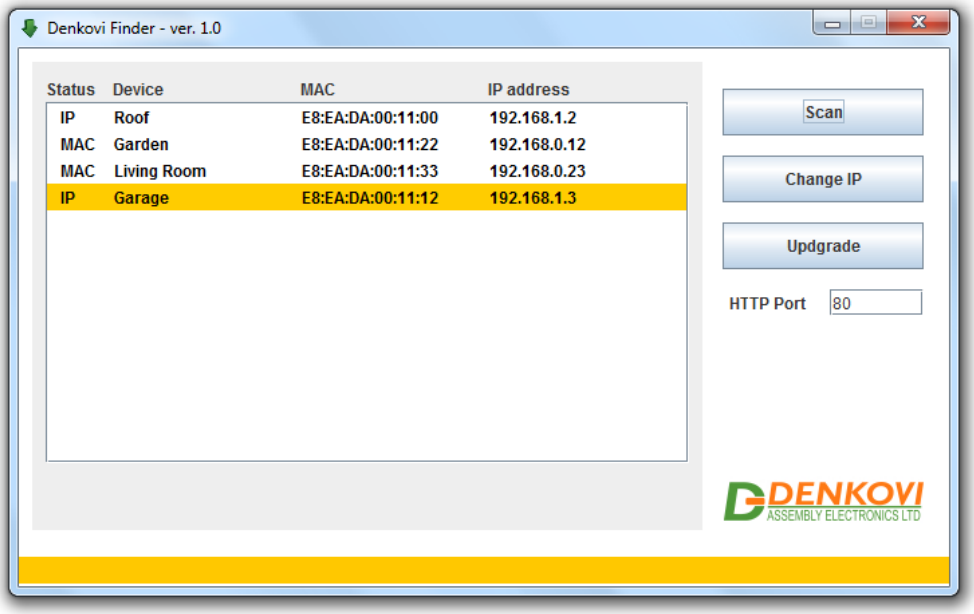

Figure 22. Denkovi Finder

6. Reset and default settings

6.1. Table with default settings

The **DAEnetIP4** module is shipped with default (factory) settings shown in Table 8. The default settings can be reloaded, if necessary (see **point 6.2**).

Table 8. Default settings

Settings group	Parameter (according Web pages)	Value
General Settings	Device Name	DAENETIP4
	Save Outputs	No
	Password	admin
Network Settings	DHCP	Disabled
	IP Address	192.168.1.100
	Gateway	192.168.1.1
	Subnet Mask	255.255.255.0
	Primary DNS	192.168.1.1
	Secondary DNS	0.0.0.0
HTTP & XML Access	HTTP Port	80
	Access IP Address	192.168.1.0
	Access Mask	0.0.0.0
	Access MAC Address	00:00:00:00:00:00
	Session Timeout, min	3
	Enable XML Access	Yes
	Encrypt XML Password	No
	Multiple XML Access	Yes
SNMP Agent	Enable SNMP	Yes
	SNMP Port	161
	Read-only Community1	public
	Read-only Community2	read
	Read-write Community1	private
	Read-write Community2	write

SNMP Traps	Enable Trap	No
	Trap sending	Level Triggered
Digital Inputs (**i** is the input number: 1-8)	Description **i**	DI**i**
	Counter **i**	0
	Filter **i**	0
	SNMP Traps **i**	Disabled
	SNMP Trap Value **i**	0 (Closed)
Analog Inputs (**i** is the input number: 1-8)	Description **i**	AI**i**
	Trap Low Treshold **i**	0
	Trap High Treshold **i**	0
	SNMP Trap **i**	Disabled
Digital Outputs (**i** is the output number: 1-16)	Description **i**	DO**i**
	Working Mode **i**	Multiple
	DI No **i**	- (no attached input)
	AI+ No **i**	- (no attached input)
	AI- No **i**	- (no attached input)
	DI No **i**	- (no attached input)
	AI Treshold 1	0
	AI Treshold 2	0
	Pulse, ms (x100)	0
	Disable Week Schedule	No (enabled)
	Use Filter	disabled
PWM Outputs (**i** is the output number: 1-2)	Description **i**	PWM**i**

6.2. Steps for loading default settings

When necessary, the factory (default settings) may be applied so the **DAEnetIP4** parameters will be returned back as those in **point 6.1** from the current document.

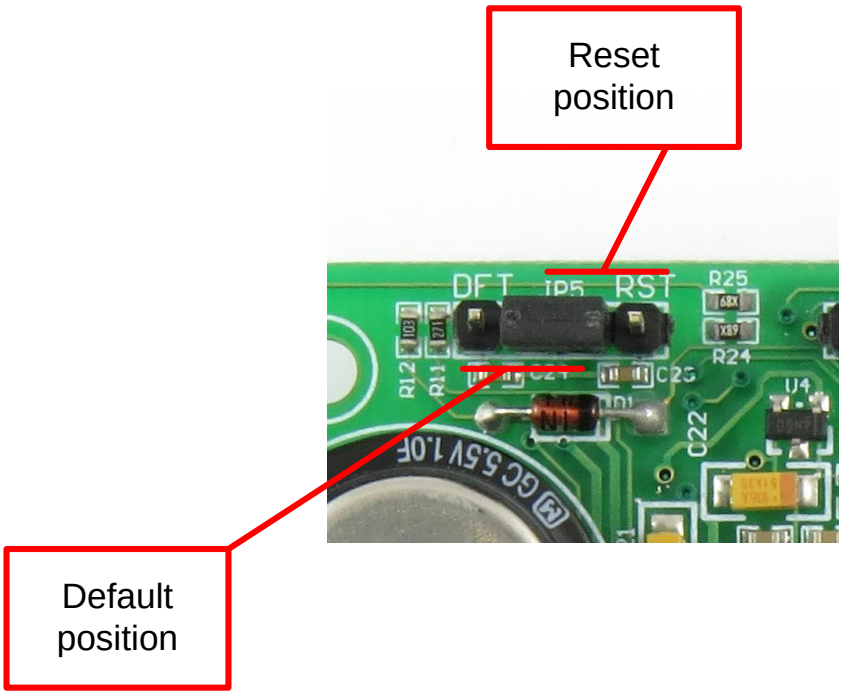

Figure 23. Loading the default settings

When **DAEnetIP4** is shipped from the factory, the jumper is placed on JP5 pins 2 and 3.

1. Turn off the power supply of the device;
2. Move the jumper to the **DFT** position (between pin 1 and 2);
3. Turn on the power supply of the device;
4. Wait until the status led become ON (approximately 10 sec);
5. Remove the jumper from the **DFT** position;
6. Turn off the power supply of the device;
7. Move back the jumper to the middle position (between pin 2 and 3);
8. The module is configured with default settings.

6.3. Restart the module

The controller may be restarted via one of the ways described bellow:

- Unplug the power supply, wait 10 seconds and plug it again;
- Move the jumper to RST position (Figure...), wait 10 seconds and then get it back to it's old position. This option is most suitable when the controller is embedded in larger system and the JP5 jumper must be extended with buttons or switches.

7. Web access

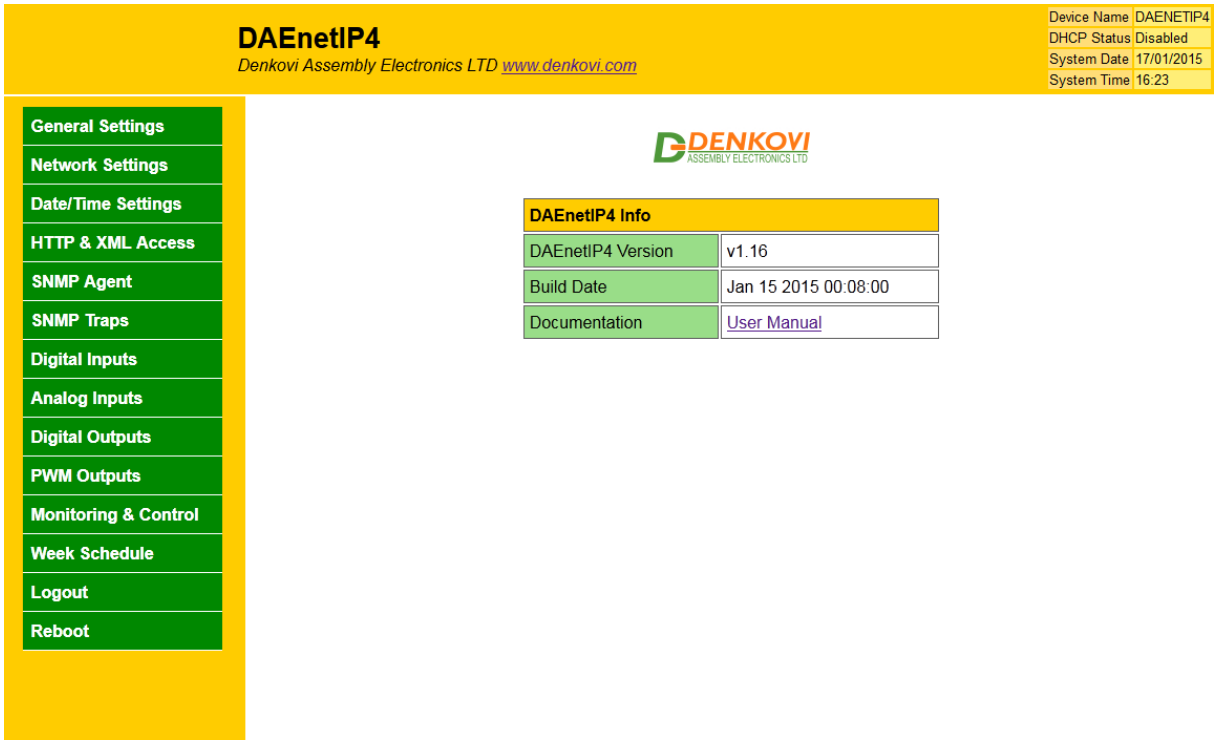

Figure 24. Web access

To access the setup pages, run a web browser (Internet Explorer, Mozilla Firefox or similar), and enter the **DAEnetIP4** IP address , for example: http://192.168.1.100

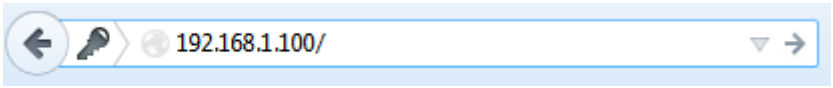

Figure 25. Open via browser

Note: You will need to have *JavaScript* enabled in your browser.

7.1. Login

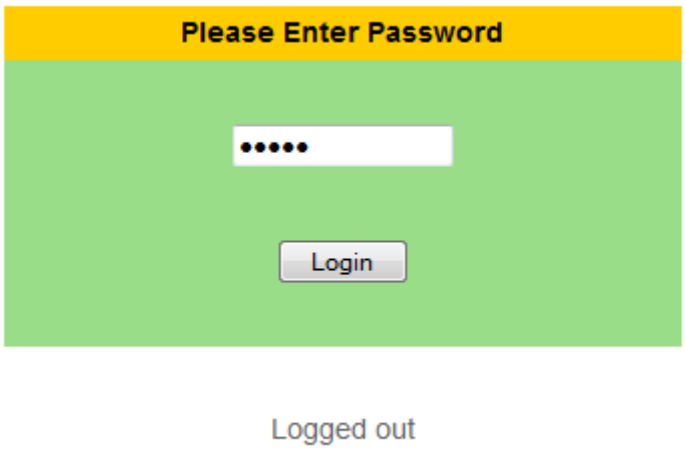

Figure 26. Login page

Enter the password and click "Login" button. This will bring you to the **DAEnetIP4** main configuration page which contains details for the current firmware version and build date and provides buttons and links to obtain further details.

Note: The default password is *admin* (passwords are case sensitive).

Note: When the password is entered, it is transmitted across the network in encrypted form, so eavesdropping on the data transmission will not reveal the password.

Note: In order to prevent setup/control conflicts, at any given moment, only one user can be logged in.

Note: If there is no data traffic between the Web-browser and the **DAEnetIP4** for time, specified by ***Session Timeout*** parameter, the session "times out" and a new login is required.

7.2. Menu

The main menu consists of the following items, located in the left window frame:

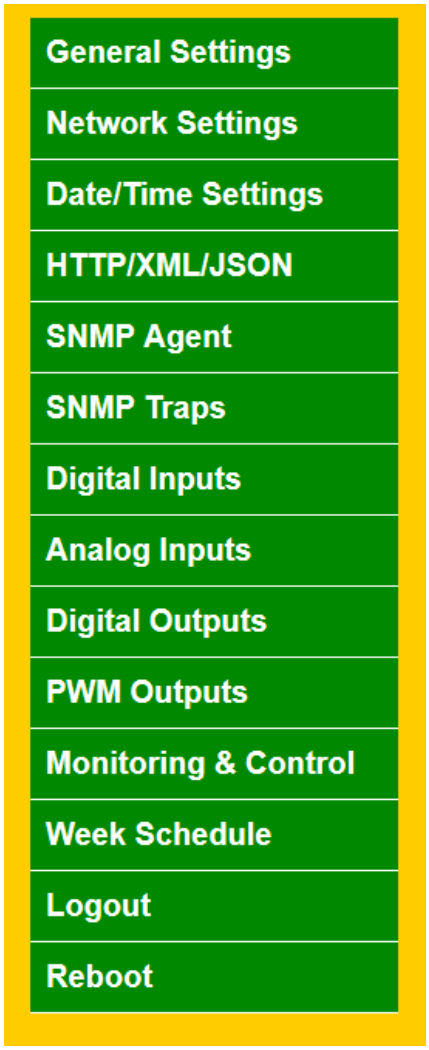

Figure 27. Navigation menu

7.3. General Settings

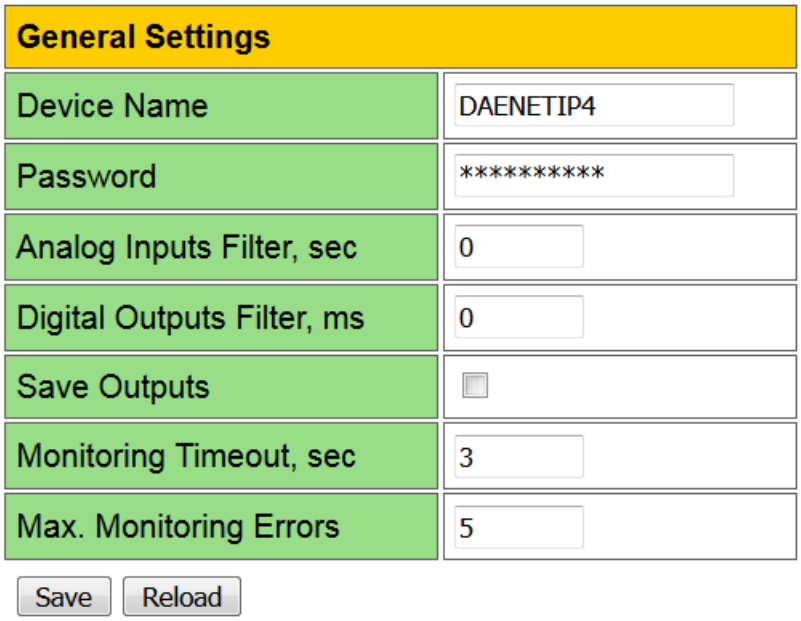

Figure 28. General settings

- **Device Name:** The name of the module (max 15 symbols). Every module can have different name in your network so they can be distinguished;
- **Password** - the password used for logging into the web admin and XML/JSON operation (max. 10 chars);

When typed, the password in this screen is not hidden. Only in this case, when the password is being changed, it is transmitted across the network "in the open". Therefore, set passwords in a secure environment where you can make sure that no one is "eavesdropping". Subsequent transmissions of the password to "login" onto the device are encrypted and "safe".

- **Analog Inputs Filter, sec** – analog inputs filter constant. The range is from 0 up to 30 sec. The filter is disabled when its value is set to 0;

This parameter sets a low pass software filter that removes the short-term fluctuations from the input signal and reduces the effect of occasional spikes. Note that the higher filter constants give a slower response to changes.

- **Digital Outputs Filter, ms** – outputs filter constant. The range is from 0 up to 9999 milliseconds. Zero value disables the filter;

In real applications, switching external loads can produce spikes and noise transients that may disturb the Analog Inputs measurement. To avoid the processing of “false” Analog Inputs measurements during these transients the parameter Relays Filter can be set to appropriate value. When a Relay changes its state, during the defined period Analog Input measurements will be ignored. The filter can be enabled or disabled individually for each relay by Use **Filter** option in Digital Outputs page.

- **Save Outputs:** When checked, each time the relays state is changed, it will be saved in non-volatile memory (EEPROM), so after reboot/restart it will be restored;
- **Monitoring Timeout, sec:** The connection timeout in seconds for the web browser;
- **Max. Monitoring Errors:** The number of sequential retries to reconnect if there is connection timeout before give up (the web browser).

This option should be used with care in dynamic systems because of restriction in maximum write cycles of the EEPROM (usually 100 000 write/erase cycles).

- **Save button:** Once you have changed the settings as required, click this button.

7.4. Network settings

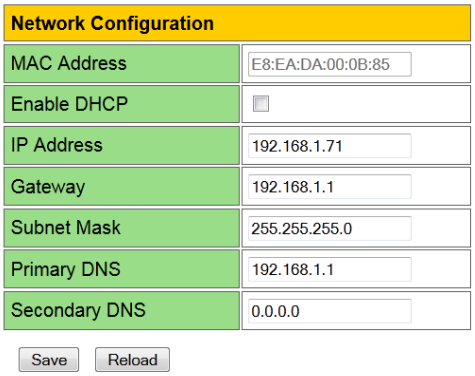

Figure 29. Network settings

This menu lets you configure the network settings of **DAEnetIP4** relay module:

- **Enable DHCP:** This option allows DHCP to be enabled or disabled. If DHCP is set to Enabled, the Network page must be saved and **DAEnetIP4** must be rebooted before obtaining an IP address;
- **IP address:** This is the IP address of the **DAEnetIP4**. It needs to be manually assigned only if DHCP is disabled. With DHCP enabled, this field displays the currently assigned address;
- **Gateway:** This specifies the IP address of the gateway router. It is used for accessing public time servers for automatic time synchronization;
- **Subnet Mask:** This is the subnet mask for the network on which the **DAEnetIP4** is installed;
- **Primary DNS:** Primary DNS (Domain Name Service) address;
- **Secondary DNS:** Secondary DNS address;
- **Save button:** Once you have changed the settings as required, click this button.

You have to reboot the device for these settings to apply.

7.5. Date and Time Settings

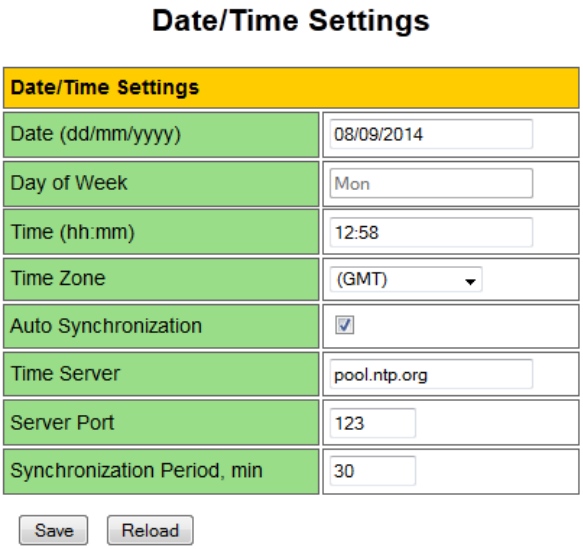

Figure 30. Date/Time settings

This page lets you configure the following parameters related with the real time clock built-in the module:

- **Date (dd/mm/yyyy):** Enter the current date here in specified format;
- **Time (hh:mm):** Enter the current time here in 24-hour format;
- **Time Zone:** Select the time zone for your geographic location.
- **Auto Synchronization:** This option enables or disables automatic synchronization with the SNTP (Simple Network Time Protocol) server with period specified by **Synchronization Period**;
- **Time Sever:** This is the SNTP server, used for synchronizing the time automatically;
- **Server Port:** SNTP server port;
- **Synchronization Period, min:** This option sets the period in which automatic synchronization will take place, if enabled;
- **Save button:** Once you have changed the settings as needed, click "**Save**". These settings apply immediately and do not require a reboot.

7.6. HTTP, XML and JSON Access

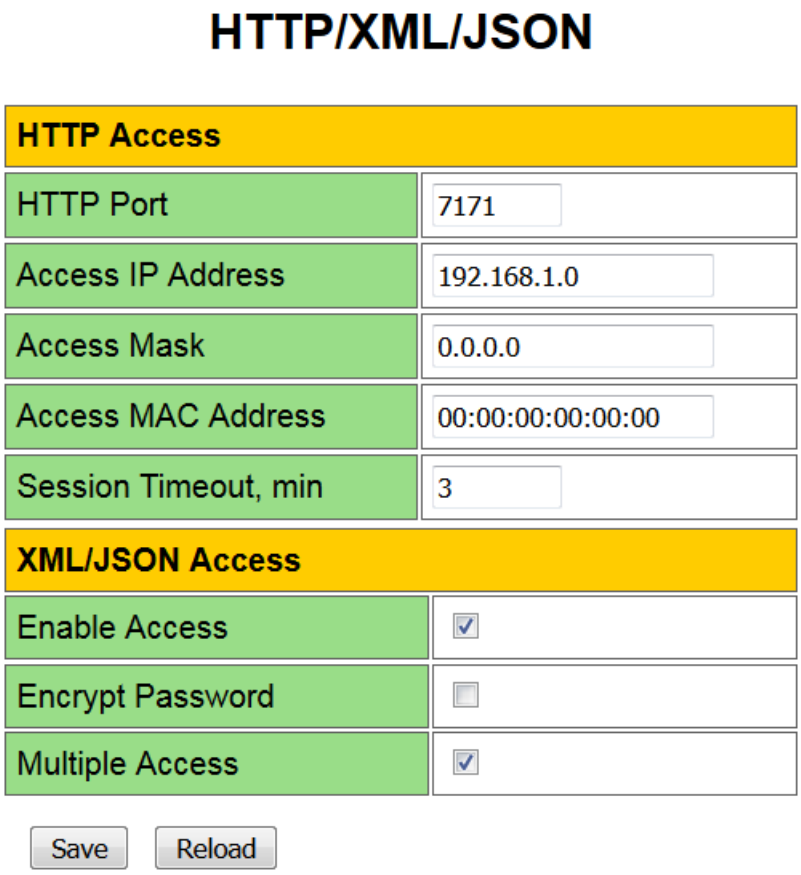

Figure 31. HTTP & XML/JSON Access

These settings let you configure the HTTP and XML/JSON access parameters of **DAEnetIP4:**

- **HTTP Port:** Port that the Web server listens for HTTP requests (default port is 80). You have to reboot the device for a new port setting to apply;
- **Access IP Address/Access Mask:** These fields can be used to restrict the HTTP/XML/JSON access by specifying the IP address and subnet mask of the HTTP/JSON client;
- **Access MAC Address:** This field can be used to restrict the HTTP/XML access by specifying the MAC address of the HTTP client;
- **Session Timeout, min:** Specifies the timeout period for HTTP and XML/JSON sessions in minutes;
- **Enable Access:** This option enables or disables XML/JSON access to the **DAEnetIP4**;
- **Encrypt Password:** When XML/JSON access is enabled, this option adds additional security level by encrypting the login password;
- **Multiple Access:** This option enables simultaneous access from several HTTP clients;

- **Save button:** Once you have changed the settings as required, click this button.

Note: When **Encrypt XML Password** mode is enabled, the **Multiple XML Access** option is not taken into account and, at any given moment, only one user can be logged-in.

Note: When **Multiple XML Access** mode is enabled, any XML request will always reset the current HTTP session.

Note: When **Multiple XML Access** mode is disabled, whether **Encrypt XML Password** is enabled or not, it is possible to access the module via XML only after login for the specified session timeout.


You have to reboot the device for these settings to apply.


It is highly recommended to log out from the web server after finishing the parameters setup.


If you don't want to restrict the HTTP/XML/JSON access by IP address, set the **Access Mask** to 0.0.0.0.


If you don't want to restrict the HTTP/XML/JSON access by MAC address, set the MAC **Address** to 00:00:00:00:00:00.


Setting the **Access Mask** to 255.255.255.255 allows the HTTP/XML/JSON access only from the exactly specified **Access IP Address**.


You can allow the HTTP/XML/JSON access to a range of IP addresses by setting an appropriate value for **Access Mask**. For example setting the **Access IP Address** to 192.168.1.0 and **Access Mask** to 255.255.255.0 allows the access from IP addresses in range from 192.168.1.0 to 192.168.1.255.

7.7. SNMP Agent

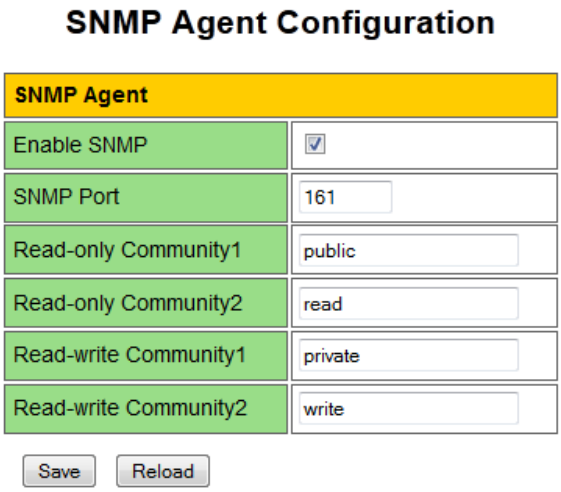

Figure 32. SNMP settings

These settings let you configure the SNMPv1 (Simple Network Management Protocol Version 1) access to the **DAEnetIP4:**

- **Enable SNMP:** This option enables or disables SNMP access to the **DAEnetIP4**;
- **SNMP Port:** UDP port number the SNMP agent receives requests on (default port is 161);
- **Read-only Community1/2:** Community string for client's authentication, used in read operations;
- **Read-write Community1/2:** Community string for client's authentication, used in read/write operations.
- **Save button:** Once you have changed the settings as required, click this button.

You have to reboot the device for these settings to apply.

7.8. SNMP Traps

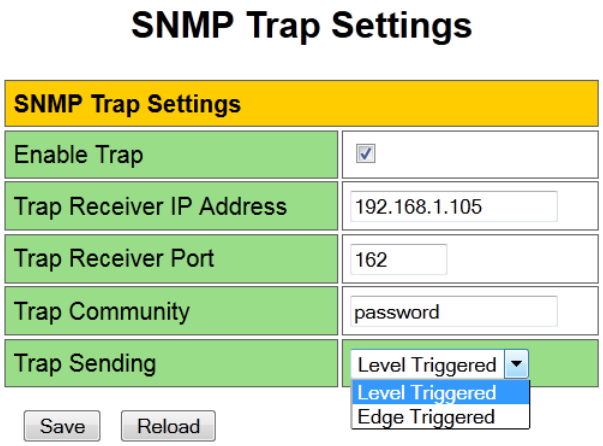

Figure 33. SNMP Trap settings

DAEnetIP4 can send SNMPv1 traps upon input event (detected by a Digital and/or Analog Input) to a Trap server and its parameters can be set from this web page:

- **Enable Trap**: Enables or disables sending traps to the server;
- **Trap Receiver IP Address**: The IP address of the trap server;
- **Trap Receiver Port:** Determines the UDP port the trap message will be sent to;
- **Trap Community**: The trap community;
- **Trap Sending:** Determines if the traps will be sent by Level or by Edge.

If it is selected the option "Level Triggered", the trap message will be sent every 5 seconds when the input is in position to generate traps. This is because the SNMP traps are UDP messages and UDP protocol does not guarantee that the packet will be received at all.

If it is selected the option "Edge Triggered", the trap message will be sent upon input event only once.

If the traps are enabled, on boot is send so called "cold trap" message indicating **DAEnetIP4** is started.

7.9. Digital Inputs

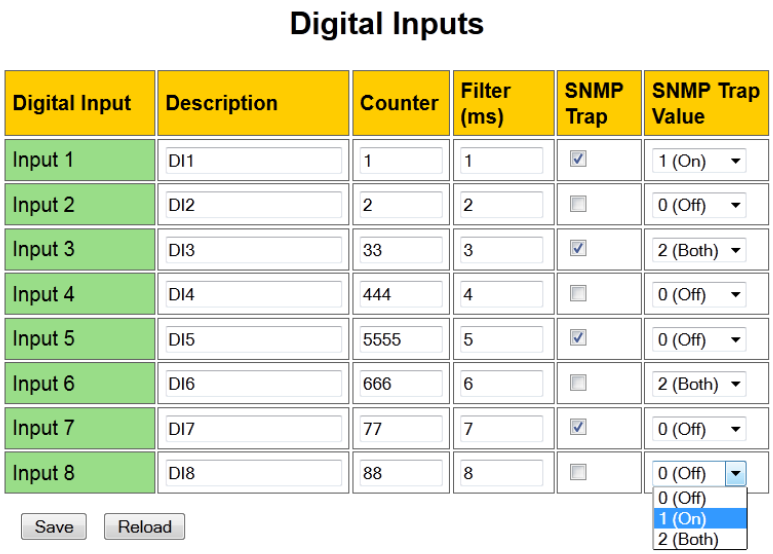

Figure 34. Digital Inputs settings

- **Description**: Digital Input identification string (max 7 chars);
- **Counter**: Every digital input works as a 16 bit counter as well. The counter is incremented at rising, falling, or both edges depending on the SNMP Trap Value. The counter is cyclic and it can be set or cleared by the user anytime via this parameter. The values which can be for this parameters are from 0 up to 65535;
- **Filter (ms)**: The input may be adjusted to work with a digital filter. It is valid for the input visualization, counting, SNMP traps and controlling the outputs as well. This parameter sets the time (in milliseconds) for this filter and it can be from 0 up to 200 ms;
- **SNMP Trap**: Determines if the Digital Input will set traps or not;
- **SNMP Trap Value:** This parameter determines how the traps will be sent:
  - *0 (Off)* - If the parameter **Trap Sending** is set with value "Level Triggered" then the trap message is sent when the Digital Input is in low level (0) every 5 seconds. If the parameter **Trap Sending** is set with value "Edge Triggered" then the trap message is sent when the Digital Input's is falling edge (1->0);
  - *1 (On)* - If the parameter **Trap Sending** is set with value "Level Triggered" then the trap message is sent when the Digital Input is in high level (1) every 5 seconds. If the parameter **Trap Sending** is set with value "Edge Triggered" then the trap message is sent during the Digital Input's rising edge (0->1);
  - 2 (Both) - If the parameter **Trap Sending** is set with value "Level Triggered" then the trap message is sent constantly every 5 seconds. If the parameter **Trap Sending** is set with value "Edge Triggered" then the trap message is sent every time when the Input changes its state (falling and rising edge).

7.10. Analog Inputs

Analog Inputs

Analog Input	Description	Min	Max	Label	Trap Low Threshold	Trap High Threshold	SNMP Trap
Input 1	Temp	0.0	10.0	V	0	0	☐
Input 2	Salon	0.0	1000.0	degC	0	0	☐
Input 3	Temp	0.0	50.0	ms2	0	0	☐
Input 4	Dupa	-29.0	297.0	%RH	0	0	☐
Input 5	Sharp	0.0	10.0	Volt	0	0	☐
Input 6	AI6	0.0	10.0	Volt	0	0	☐
Input 7	AI7	0.0	10.0	Volt	0	0	☐
Input 8	AI8	0.0	10.0	Volt	0	0	☐

Save Reload

Figure 35. Analog Inputs settings

- **Description**: Analog Input identification string (max 7 chars);
- **Min** - the value used for scaling where the analog input is with value 0 (minimum value: -9999.9, maximum value: 9999.9);
- **Max** - the value used for scaling where the analog input is with value 1024 (minimum value: -9999.9, maximum value: 9999.9);
- **Label** - the units for the scaled value for example: m,A,KG,V...(max 4 chars);
- **Low Threshold** - the low threshold limit for this input (from -9999 up to 9999);
- **High Threshold** - the high threshold limit for this input (from -9999 up to 9999);
- **SNMP Trap**: Determines if the Analog Input will set traps or not;

The scaled value for the analog inputs is calculated in the following way:

$$ScaledADCvalue = Min + \frac{Max - Min}{1024} * PureADCValue$$

7.11. Digital Outputs

Digital Outputs Settings

Digital Output	Description	Working Mode	DI No	DI Mode	AI+ No	AI- No	AI Threshold 1	AI Threshold 2	Pulse, ms (x100)	Disable Week Schedule	Use Filter
Output 1	DO1	Multiple	0	Normal	0	0	0	0	0		
Output 2	DO2	Multiple	0	Normal	0	0	0	0	0		
Output 3	DO3	Multiple	0	Normal	0	0	0	0	0		
Output 4	Boiler	Multiple	0	Normal	0	0	0	0	0		
Output 5	warm_Wa	Multiple	0	Normal	0	0	0	0	0		
Output 6	Kalt_Wa	Multiple	0	Normal	0	0	0	0	0		
Output 7	DO7	Multiple	0	Normal	0	0	0	0	0		
Output 8	DO8	Multiple	0	Normal	0	0	0	0	0		
Output 9	DO9	Multiple	0	Normal	0	0	0	0	0		
Output 10	DO10	Multiple	0	Normal	0	0	0	0	0		
Output 11	DO11	Multiple	0	Normal	0	0	0	0	0		
Output 12	DO12	Multiple	0	Normal	0	0	0	0	0		
Output 13	DO13	Multiple	0	Normal	0	0	0	0	0		
Output 14	DO14	Multiple	0	Normal	0	0	0	0	0		
Output 15	DO15	Multiple	0	Normal	0	0	0	0	0		
Output 16	DO16	Multiple	0	Normal	0	0	0	0	0		

Save Reload

Figure 36. Digital Output names

- **Description**: Output identification string (max 7 chars);

This description will appear in XML/JSON files, as well as in the Monitoring & Control page.

- **Working Mode**: Determines how to work the Digital Output:
  - *Regulator* - in this mode the Output can be controlled only by Analog Input. This mode is designed for cases where the input event depends on the output reaction (like thermo regulator for example). In this mode the parameters "DI No" and "Pulse" are not taken in mind. Also in this mode it is not possible to control the output via SNMP, Web browser or HTTP/XML protocol.
  - *Multiple* - in this mode the output can be controlled by Analog Inputs, Digital Inputs, week schedule or manual control (via browser, SNMP, HTTP/XML). This mode is designed for cases where the input event does not depend on the output reaction (for example controlling lamp depending on the sunlight);
- **DI No**: The number of the Digital Input which is "attached" to this output. It can be "-" if there is no Digital Input attached or any number between 1 and 8. If there is a selected input here and the working mode is "Multiple", then the output will be in high level when the input is in high level and will be in low level when the input is in low level;
- **DI Mode** – one of eight level/edge control modes can be selected:
  - *Normal* - the Relay state is On when the Digital Input state is 1 (On) and Off when the Digital Input state is 0 (Off);
  - *Inverse* - the Relay state is On when the Digital Input state is 0 (Off) and Off when the Digital Input state is 1 (On);

- *Toggle(DI=0)* - a falling edge (1 -> 0) of the Digital Input toggles the Relay state between On and Off;
- *Toggle(DI=1)* - a rising edge (0 -> 1) of the Digital Input toggles the Relay state between On and Off;
- *Normal(DI=0)* - a falling edge (1 -> 0) of the Digital Input switches the Relay Off;
- *Normal(DI=1)* - a rising edge (0 -> 1) of the Digital Input switches the Relay On;
- *Inverse(DI=0)* - a falling edge (1 -> 0) of the Digital Input switches the Relay On;
- *Inverse(DI=1)* - a rising edge (0 -> 1) of the Digital Input switches the Relay Off.

- **AI+ No** - the number of the Analog Input (1 to 8) which is "attached" to this output. If AI- No is set to zero, this is the single-ended input, otherwise – the (+) lead of the differential input.
- **AI- No** – if set to non-zero (1 to 8), this is the (-) lead of the differential input;

If **AI+ No** is non-zero and **AI- No** is zero, the output is controlled by the **AI+ No** measured value.

If both **AI+ No** and **AI- No** are non-zero, the output is controlled by the difference of **AI+ No** and **AI- No** measured values.

If **AI+ No** is zero, the output is not controlled by the Analog Input (in this case the **AI- No** is not relevant).

- **AI Treshold 1**: The first level for controlling the output (from -9999 up to 9999);
- **AI Treshold 2**: The second level for controlling the output (from -9999 up to 9999);

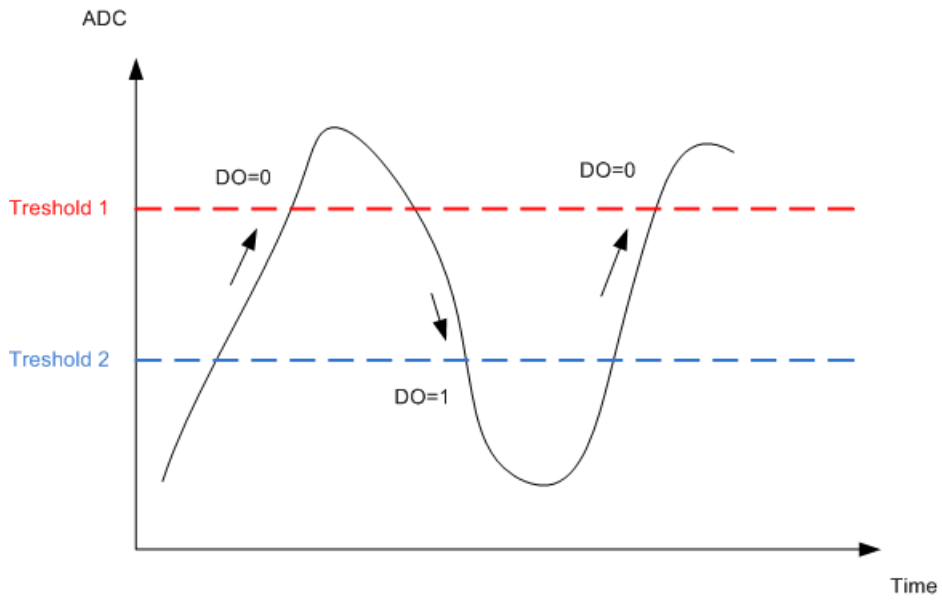

Figure 37. Treshold 1 > Treshold 2

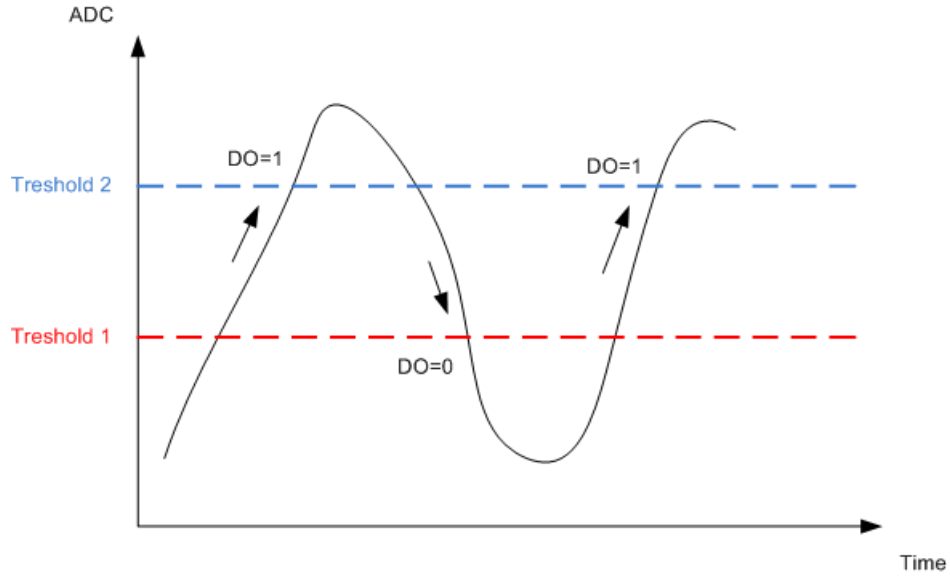

Figure 38. Treshold 1 < Treshold 2

- **Disable Week Schedule**: Enables/Disables globally the week schedule control for this output. This means the output may be in the schedule, but if this flag is enabled, then it won't be activated by the schedule;
- **Pulse, ms (x100)**: Determines if the output works in pulse mode and if so what is the duration of the single pulse (in milliseconds x 100 - for example value of 10 means 1 second). This parameter can accept values between 0 and 65535. If it is 0, then the pulse mode is disabled. If it is between 1 and 65535, then the pulse mode for this output is activated and it will be hold in high level for the specified time by this parameter. During this time, the output can be set in low via Analog Input, digital input, week schedule, SNMP, web browser control, HTTP/XML. The "pulse" mode is active only if the parameter **Working Mode** is set with value "Multiple". This means that:
  - If the output is set to high level via browser manually, SNMP or HTTP/XML it will be in high level for the determined time by this parameter and then set back to low level;
  - If the output is controlled by Analog Input and it is in "Multiple" mode, then it will be hold to high level for time specified by this parameter and then will be set to low;
  - If the output is controlled by a Digital Input, then when the input becomes in high level, the output will be set to high level for time, specified by this parameter and then will be set to low;
  - If the output is controlled by weekly schedule, then when the output becomes in high level, it will be set to high level for time, specified by this parameter and then will be set to low;
  - If the output is controlled via SNMP, web browser, HTTP/XML, then when the output becomes in high level, it will be set to high level for time, specified by this parameter and then will be set to low;

7.12. PWM Outputs

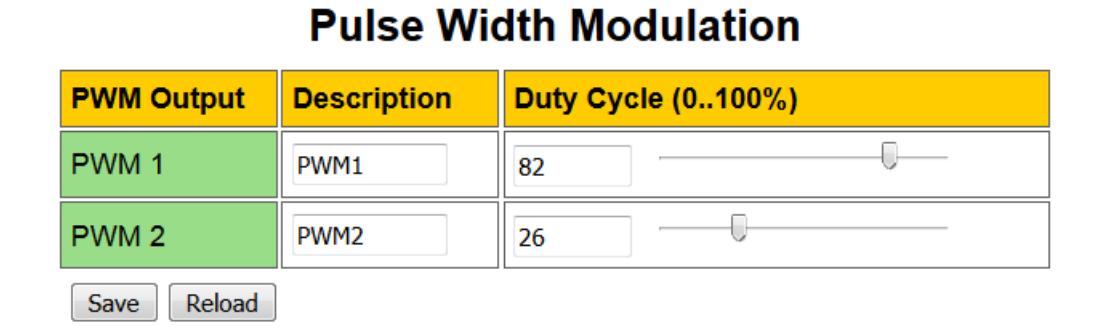

Figure 39. PWM settings

- **Description**: PWM output identification string (max 7 chars);
- **Duty Cycle (0..100%)**: Determines the duty cycle of the PWM outputs.

The PWM frequency is 40 KHz

7.13. Monitoring and control

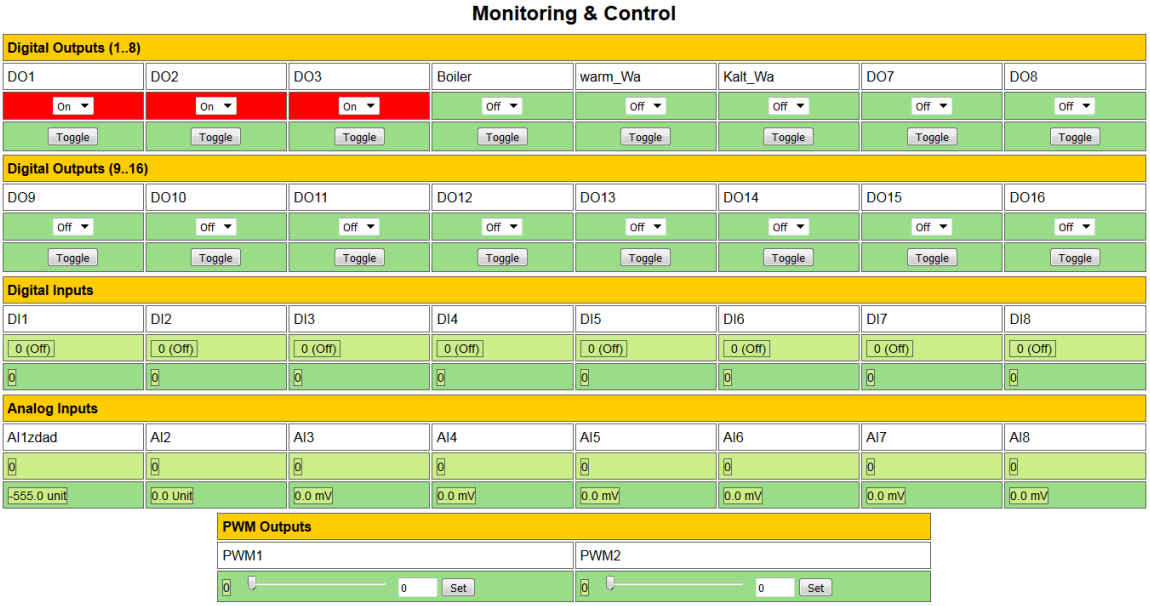

Figure 40. Monitoring and control

This page provides monitoring and control of the **DAEnetIP4** I/O via AJAX requests (almost in real time).

7.14.Week Schedule

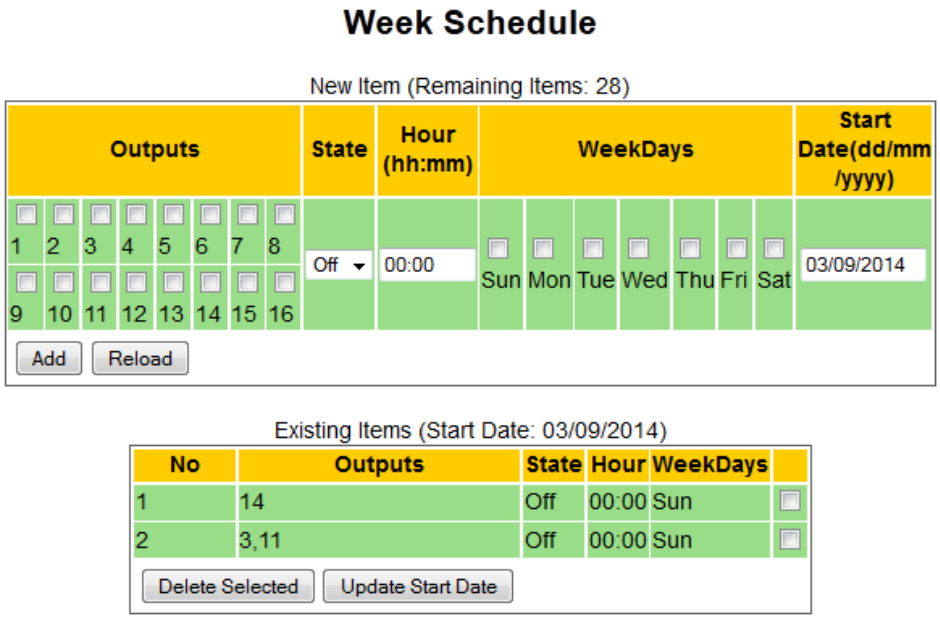

Figure 41. Week schedule

This page configures the **Week Schedule** table for switching **Digital Outputs** in High (ON) or Low (OFF) at specific times. You can add up to **30** time items to the list. The top table of this page allows you to define a new item, while the bottom table shows the already defined list:

- **Outputs:** Select a group of relays that should be switched;
- **State:** Defines the state (ON/OFF) for the selected group of outputs;
- **Hour:** Time the group of outputs will be switched at;
- **WeekDays:** Select the days the defined switching should take place;
- **Start Date (dd/mm/yyyy):** The start date for the **Week Schedule** table.

Once you have defined a new item, click "**Add**". This item will be added as a new row in a **Week Schedule** table.

This feature allows you to turn on/off specific outputs upon certain date and time or weekday without the need of LAN connection between the computer and the module.

To delete an item, select it in **Existing Items** table and click on "**Delete Selected**" button.

To set a new start date, click on "Update Start Date" button.

The module has back-up supply source for the RTC in order to keep the current date/time for several days during power off.

7.15. Logout

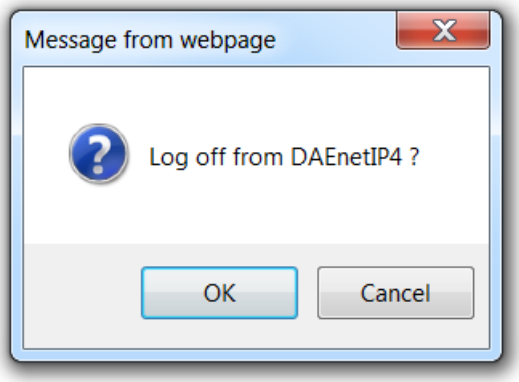

Figure 42. Log off

7.16. Reboot

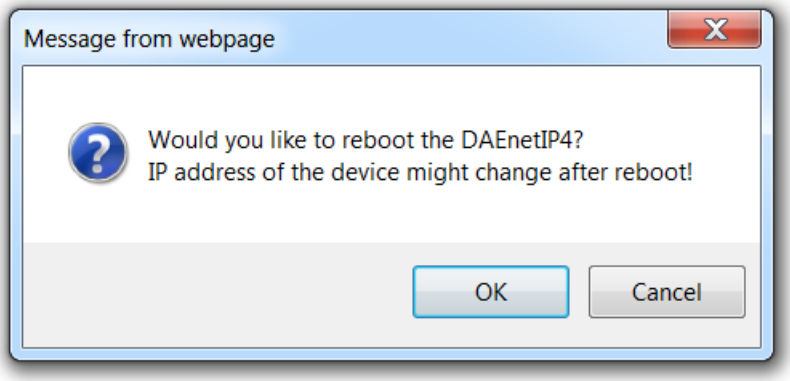

Figure 43. Reboot

8. HTTP/XML/JSON access

This operation mode allows custom applications to control the **DAEnetIP4** without using a Web-browser. The custom application acts as a HTTP client, sending HTTP GET requests to the **DAEnetIP4**.

To receive the current state of the **DAEnetIP4,** the application requests the page *current_state.xml*, for example:

http://192.168.1.100/current_state.xml

The custom application can also control the **DAEnetIP4** by sending parameters (name/value pairs) with the HTTP request, for example:

http://192.168.1.100/current_state.xml?pw=admin&Output1=1&PWM2=23&Output16=0&Pulse5=20&Count2=0

The above command makes the following:

- Set Digital Output #1 in high level;
- Set PWM #2 to 23%;
- Set Digital Output #16 in low level;
- Makes single pulse to Digital Output 5 with length 2 seconds;
- Clears the counter of Digital Input 2.

The XML login process differs depending on the selected **Encrypt XML Password** option.

8.1. Login (Encrypted Password)

In this mode a two-step login sequence is provided as a protection against unauthorized access. The first time the custom application requests the page *current_state.xml / current_state.json*, a random login key is issued in the reply. Next the custom application uses this key to encrypt the password. The encrypted password is sent as a parameter with the next request to the page *current_state.xml / current_state.json*.

Bellow is an example of login process:

Step 1:
Request
http://192.168.1.100/current_state.xml
Reply (login required):

```
<CurrentState>
<LoginKey>65156</LoginKey>
</CurrentState>
```

Request
http://192.168.1.100/current_state.json
Reply (login required):
{
"CurrentState": {"LoginKey": "65156"}
}

Step 2:
Request (password is sent as a parameter)
http://192.168.1.100/current_state.xml?pw=28237099263eabfd88626124a822c64c

Reply (password is O'K, login accepted):
<CurrentState>
…
</CurrentState>

Request (password is sent as a parameter)
http://192.168.1.100/current_state.json?pw=28237099263eabfd88626124a822c64c

Reply (password is O'K, login accepted):
{
"CurrentState": {
… }
}

Password encryption algorithm to be implemented in custom application is available upon request.

8.2. Login (Non-Encrypted Password)

In this mode the password is passed as non-encrypted parameter with the request:

http://192.168.1.100/current_state.xml?pw=admin

http://192.168.1.100/current_state.json?pw=admin

Getting the <LoginKey> in the answer in this mode means only that the provided password is wrong or the login session has been expired.

If there is no data traffic between the custom application and the **DAEnetIP4** for time, specified by **Session Timeout** parameter, the session "times out" and a new login is required.

8.3. Getting the I/O current state

After a login the custom application can obtain the **DAEnetIP4** current state by a request to the page ***current_state.xml*** or ***current_state.json*** with or without login procedure (depending on the settings):

http://192.168.1.100/current_state.xml

The reply contains page in XML format:


```
▼<CurrentState>
  ▼<DigitalInput1>
      <Name>Sensor1</Name>
      <Value>0</Value>
      <Count>11</Count>
    </DigitalInput1>
  ▶<DigitalInput2>...</DigitalInput2>
  ▶<DigitalInput3>...</DigitalInput3>
  ▶<DigitalInput4>...</DigitalInput4>
  ▶<DigitalInput5>...</DigitalInput5>
  ▶<DigitalInput6>...</DigitalInput6>
  ▶<DigitalInput7>...</DigitalInput7>
  ▶<DigitalInput8>...</DigitalInput8>
  ▼<AnalogInput1>
      <Name>Lm35</Name>
      <Value>107</Value>
    </AnalogInput1>
  ▶<AnalogInput2>...</AnalogInput2>
  ▶<AnalogInput3>...</AnalogInput3>
  ▶<AnalogInput4>...</AnalogInput4>
  ▶<AnalogInput5>...</AnalogInput5>
  ▶<AnalogInput6>...</AnalogInput6>
  ▶<AnalogInput7>...</AnalogInput7>
  ▶<AnalogInput8>...</AnalogInput8>
  ▼<Output1>
      <Name>Pump</Name>
      <Value>0</Value>
    </Output1>
  ▶<Output2>...</Output2>
  ▶<Output3>...</Output3>
  ▶<Output4>...</Output4>
  ▶<Output5>...</Output5>
  ▶<Output6>...</Output6>
  ▶<Output7>...</Output7>
  ▶<Output8>...</Output8>
  ▶<Output9>...</Output9>
  ▶<Output10>...</Output10>
  ▶<Output11>...</Output11>
  ▶<Output12>...</Output12>
  ▶<Output13>...</Output13>
  ▶<Output14>...</Output14>
  ▶<Output15>...</Output15>
  ▶<Output16>...</Output16>
  ▼<PWM1>
      <Name>PWM1</Name>
      <Value>40</Value>
    </PWM1>
  ▶<PWM2>...</PWM2>
</CurrentState>
```

http://192.168.1.100/current_state.json

The reply contains page in JSON format:


```
{
  "CurrentState": {
    "DigitalInput": [
      {"Name": "Ho", "Value": "0", "Count": "0"},
      {"Name": "DI2", "Value": "0", "Count": "0"},
      {"Name": "DI3", "Value": "0", "Count": "0"},
      {"Name": "DI4", "Value": "0", "Count": "0"},
      {"Name": "DI5", "Value": "0", "Count": "14"},
      {"Name": "DI6", "Value": "0", "Count": "0"},
      {"Name": "DI7", "Value": "0", "Count": "0"},
      {"Name": "DI8", "Value": "0", "Count": "0"}
    ],
    "AnalogInput": [
      {"Name": "Temp", "Value": "488", "Measure": "4.7 V"},
      {"Name": "Salon", "Value": "22", "Measure": "21.5 degC"},
      {"Name": "Temp", "Value": "2", "Measure": "0.1 ms2"},
      {"Name": "Dupa", "Value": "349", "Measure": "82.2 %RH"},
      {"Name": "Sharp", "Value": "102", "Measure": "0.9 Volt"},
      {"Name": "AI6", "Value": "1", "Measure": "0.0 Volt"},
      {"Name": "AI7", "Value": "1", "Measure": "0.0 Volt"},
      {"Name": "AI8", "Value": "1", "Measure": "0.0 Volt"}
    ],
    "Output": [
      {"Name": "dre", "Value": "0"},
      {"Name": "rti", "Value": "0"},
      {"Name": "DO.3", "Value": "0"},
      {"Name": "D0.4", "Value": "0"},
      {"Name": "Salon", "Value": "0"},
      {"Name": "DO.6", "Value": "0"},
      {"Name": "room", "Value": "1"},
      {"Name": "yut", "Value": "1"},
      {"Name": "Bel", "Value": "1"},
      {"Name": "DO.10", "Value": "1"},
      {"Name": "DO.11", "Value": "0"},
      {"Name": "DO.12", "Value": "0"},
      {"Name": "DO.13", "Value": "0"},
      {"Name": "DO.14", "Value": "0"},
      {"Name": "DO.15", "Value": "0"},
      {"Name": "DO.16", "Value": "0"}
    ],
    "PWM": [
      {"Name": "Light 1", "Value": "0"},
      {"Name": "PWM2", "Value": "0"}
    ]
  }
}
```

8.4. Multiple XML/JSON Access

In this mode the password should be passed as non-encrypted parameter with each request:

http://192.168.1.100/current_state.xml?pw=admin&Output1=1&PWM2=23&Output16=0&Pulse5=20&Count2=0

http://192.168.1.100/current_state.json?pw=admin&Output1=1&PWM2=23&Output16=0&Pulse5=20&Count2=0

Multiple XML/JSON Access is not allowed when **Encrypt XML/JSON Password** option is enabled.

8.5. Parameters

After a login the custom application can also control the **DAEnetIP4** by sending parameters (name/value pairs) with the HTTP request.

Valid parameters and values are shown in the bellow table.

Table 9. Valid HTTP parameters

Name	Value	Description
Outputi Di (from v1.25)	0	Switch the Output Off
	1	Switch the Output ON
	2	Toggle the Output state
PWMi Wi (from v1.25)	0...100	PWM Outputi value (i=1...2)
Pulsei Pi (from v1.25)	1...65535	Generate pulse to an output (i=1...16)
Counti Ci (from v1.25)	0...65535	Set the counter value for a digital input (i=1...8)
SetAll	0...65535	Set all the digital outputs via single parameter
pw	password	Required at login

Please note from firmware v1.25 the HTTP GET process buffer is decreased from 117 to 101 bytes. This affects to the HTTP GET requests parameters maximum length. An additional attention must be taken to existing applications using especially long HTTP GET requests. Also please note from v1.25 they are implemented and short commands (Table 9).

9. SNMP access

DAEnetIP4 supports SNMPv2 protocol. Most of the parameters can be configured/read via snmp commands. Read-only community string is used for reading and Read-Write Community String is used for changing the parameters. Parameters that can be changed, are grouped according to their functions in the tables below. To obtain a valid OID number it is necessary to replace the "x" symbol with the prefix "**.1.3.6.1.4.1.42505**". Also all the SNMP commands are described in the MIB file**.**

9.1. Product

Table 10. Product parameters

OID	Name	Access	Description	Syntax
x.1.1.1.0	Name	read-only	Description of the module	DISPLAYSTRING
x.1.1.2.0	Version	read-only	Current firmware version	DISPLAYSTRING
x.1.1.3.0	Date	read-only	Current firmware version build date	DISPLAYSTRING

9.2. Traps

Table 11. Traps parameters

Start OID	Name	Access	Description	Syntax
x.1.4.1.0	TrapEnabled	read-write	Trap Enable Flag (Disabled-0, Enabled-1)	INTEGER { no(0),yes(1) }
x.1.4.2.0	TrapReceiverIPAddress	read-write	Trap Receiver IP Address	IPADDRESS
x.1.4.3.0	TrapReceiverPort	read-write	Trap Receiver Port (0..65535)	INTEGER32 (0..65535)
x.1.4.4.0	TrapCommunity	read-write	Community in Sending Trap	DISPLAYSTRING (SIZE (0..7))
x.1.4.5.0	TrapSending	read-write	Trap Sending (Level Triggered-0, Edge Triggered-1)	INTEGER {(0),(1)}

9.3. Setup

Table 12. Digital Outputs OID-s

Start OID	End OID	Name	Access	Description	Syntax
x.1.2.3.1.2.0	x.1.2.3.1.2.15	DigitalOutputDescription	read-write	Digital Output Description (maxlen=7)	DISPLAYSTRING (SIZE (0..7))
x.1.2.3.1.3.0	x.1.2.3.1.3.15	DigitalOutputWorkingMode	read-write	Digital Output Working Mode (Multiple-0, Regulator-1)	INTEGER { Multiple(0), Regulator(1) }

x.1.2.3.1.4.0	x.1.2.3.1.4.15	DigitalOutputAnalogPlusInputNo	read-write	Digital Output <- Analog Input (+) Control (None-0, AIn1-1, AIn2-2, ..., AIn8-8)	INTEGER (0..8)
x.1.2.3.1.5.0	x.1.2.3.1.5.15	DigitalOutputDigitalInputNo	read-write	Digital Output <- Digital Input Control (None-0, DIn1-1, DIn2-2, ..., DIn8-8)	INTEGER (0..8)
x.1.2.3.1.6.0	x.1.2.3.1.6.15	DigitalOutputAnalogTreshold1	read-write	Digital Output <- Analog Treshold 1 (0..1023)	INTEGER (0..1023)
x.1.2.3.1.7.0	x.1.2.3.1.7.15	DigitalOutputAnalogTreshold2	read-write	Digital Output <- Analog Treshold 2 (0..1023)	INTEGER (0..1023)
x.1.2.3.1.8.0	x.1.2.3.1.8.15	DigitalOutputDisableWeekSchedule	read-write	Digital Output Disable Week Schedule (off-0, on-1)	INTEGER { off(0), on(1) }
x.1.2.3.1.9.0	x.1.2.3.1.9.15	DigitalOutputSetPulsePeriod	read-write	Digital Output Set Pulse Period, ms (x100) (0..65535)	INTEGER32 (0..65535)
x.1.2.3.1.10.0	x.1.2.3.1.10.15	DigitalOutputStartPulse	read-write	Digital Output Start Pulse, ms (x100) (0..65535)	INTEGER32 (0..65535)
x.1.2.3.1.11.0	x.1.2.3.1.11.15	DigitalOutputState	read-write	Digital Output State (off-0, on-1)	INTEGER { off(0), on(1) }
x.1.2.3.1.12.0	x.1.2.3.1.12.15	DigitalOutputInputMode	read-write	Digital Output <- Digital Input Mode (Normal-0, Inverse-1, Toggle(DI=0)-2, Toggle(DI=1)-3, Normal(DI=0)-4, Normal(DI=1)-5, Inverse(DI=0)-6,	INTEGER {Normal-0, Inverse-1, Toggle(DI=0)-2, Toggle(DI=1)-3, Normal(DI=0)-4, Normal(DI=1)-5, Inverse(DI=0)-6, Inverse(DI=1)-7 }

				Inverse(DI=1)-7)	
x.1.2.3.1.13.0	x.1.2.3.1.13.15	DigitalOutputAnalogMinusInputNo	read-write	Digital Output <- Analog Input (-) Control (None-0, AIn1-1, AIn2-2, ..., AIn8-8)	INTEGER (0..8)
x.1.2.3.1.16.0	x.1.2.3.1.16.15	DigitalOutputUseFilter	read-write	Digital Output Use Filter (No-0, Yes-1)	INTEGER {No(0), Yes(1) }

Table 13. Digital Inputs OID-s

Start OID	End OID	Name	Access	Description	Syntax
x.1.2.1.1.2.0	x.1.2.1.1.2.7	DigitalInputName	read-write	Digital Input Name (maxlen=7)	DISPLAYSTRING (SIZE (0..7))
x.1.2.1.1.3.0	x.1.2.1.1.3.7	DigitalInputState	read-only	Digital Input State (closed-0,open-1)	INTEGER { closed(0),open(1) }
x.1.2.1.1.4.0	x.1.2.1.1.4.7	DigitalInputCounter	read-write	Digital Input Counter (0..65535)	INTEGER32 (0..65535)
x.1.2.1.1.5.0	x.1.2.1.1.5.7	DigitalInputFilter	read-write	Digital Input Filter (0..200 ms)	INTEGER (0..200)
x.1.2.1.1.6.0	x.1.2.1.1.6.7	DigitalInputFilter	read-write	Digital Input Trap Enable Flag	INTEGER { no(0),yes(1) }
x.1.2.1.1.7.0	x.1.2.1.1.7.7	DigitalInputTrapValue	read-write	Digital Input Trap Value (closed-0,open-1,both-2)	INTEGER { closed(0),open(1),both(2) }

Table 14. Analog Inputs OID-s

Start OID	End OID	Name	Access	Description	Syntax
x.1.2.2.1.2.0	x.1.2.2.1.2.7	AnalogInputDescription	read-write	Analog Input Description (maxlen=7)	DISPLAYSTRING (SIZE (0..7))
x.1.2.2.1.3.0	x.1.2.2.1.3.7	AnalogInputTrapLowThreshold	read-write	Analog Input Trap Low Threshold (-9999..9999)	INTEGER (-9999..9999)
x.1.2.2.1.4.0	x.1.2.2.1.4.7	AnalogInputTrapHighThreshold	read-write	Analog Input Trap High Threshold (-9999..9999)	INTEGER (-9999..9999)
x.1.2.2.1.5.0	x.1.2.2.1.5.7	AnalogInputTrapEnabled	read-write	Analog Input Trap Enable	INTEGER { no(0),yes(1)

				Flag (Disabled-0, Enabled-1)	}
x.1.2.2.1.6.0	x.1.2.2.1.6.7	AnalogInput Value	read-write	Analog Input ADC Value	INTEGER { no(0),yes(1) }
x.1.2.2.1.7.0	x.1.2.2.1.7.7	AnalogInput Min	read-write	Analog Input Min Value (maxlen=7)	DISPLAYSTRING (SIZE (0..7))
x.1.2.2.1.8.0	x.1.2.2.1.8.7	AnalogInput Max	read-write	Analog Input Min Value (maxlen=7)	DISPLAYSTRING (SIZE (0..7))
x.1.2.2.1.9.0	x.1.2.2.1.9.7	AnalogInputLabel	read-write	Analog Input Label (maxlen=4)	DISPLAYSTRING (SIZE (0..3))
x.1.2.2.1.10.0	x.1.2.2.1.10.7	AnalogInput Measure	read-write	Analog Input Measured Value	DISPLAYSTRING

Table 15. PWM Outputs OID-s

Start OID	End OID	Name	Access	Description	Syntax
x.1.2.4.1.2.0	x.1.2.4.1.2.1	PWMOutputs Description	read-write	PWM Output Description (maxlen=7)	DISPLAYSTRING (SIZE (0..7))
x.1.2.4.1.3.0	x.1.2.4.1.3.1	PWMDutyCycle	read-write	PWM Output Duty Cycle (0..100), %	INTEGER (0..100)

9.4. Week Schedule

Table 16. Week Schedule parameters

Start OID	End OID	Name	Access	Description	Syntax
x.1.5.1.0	...	WeekScheduleStartDate	read-write	Week Schedule Start Date (dd/mm/yyyy)	DISPLAYSTRING
x.1.5.2.1.2.0	x.1.5.2.1.2.29	Enabled	read-write	Week Schedule Row Enable Flag (Disabled-0, Enabled-1)	INTEGER { no(0),yes(1) }
x.1.5.2.1.3.0	x.1.5.2.1.3.29	Outputs	read-write	Outputs Code (0..65535), Output1 - bit 0, ..., Output16 - bit 15	INTEGER32 (0..65535)
x.1.5.2.1.4.0	x.1.5.2.1.4.29	OutputsState	read-write	Outputs Code (0..65535), Output1 - bit 0, ...,	Outputs State (off-0, on-1)

				Output16 - bit 15	
x.1.5.2.1.5.0	x.1.5.2.1.5.29	Hour	read-write	Hour (hh:mm)	DISPLAYSTRING
x.1.5.2.1.6.0	x.1.5.2.1.6.29	WeekDays	read-write	WeekDays Code (0..127), Sunday - bit 0, ..., Saturday - bit 6	INTEGER (0..127)

9.5. Control

Table 17. Control

OID	Name	Access	Description	Syntax
x.1.3.1.0	DigitalInputsState	read-write	Digital Inputs State (0..255)	INTEGER (0..255)
x.1.3.2.0	DigitalOutputsState	read-write	Digital Outputs State (0..65535)	INTEGER32 (0..65535)
x.1.3.3.0	SystemDate	read-write	System Date (dd/mm/yyyy)	DISPLAYSTRING
x.1.3.4.0	SystemTime	read-write	System Time (hh:mm)	DISPLAYSTRING
x.1.3.5.0	Reboot	read-write	Reboot **DAEnetIP4**	INTEGER (0..255)
x.1.3.6.0	sysUpTime	read-only	The time (in hundredths of a second) since the device was last re-initialized.	TIMETICKS
x.1.3.7.0	AnalogInputFilter	read-write	Analog Input Filter Constant (0..30), sec	INTEGER (0..30)
x.1.3.8.0	INTEGER (0..9999)	read-write	Digital Outputs Filter, ms (0..9999)	INTEGER (0..9999)

To reboot the device via SNMP, set the Reboot value to the ASCII code of the first char of your Web password. For example, if this is the char 'a', code in decimal is 97.

10. Security considerations

The **DAEnetIP4** runs a special firmware and do not have a general-purpose operating system. There are no extraneous IP services found on general-purpose operating systems (e.g. fingerd, tcp_wrapper, etc.) that can possibly be exploited by an unauthorized agent. In particular, the **DAEnetIP4** does not run protocols such as Telnet and FTP which may have the potential for security breech. The only exception from this is the SNMPv2 protocol, that can be disabled.

Web-browser access
A challenge-response authentication is used in login process. When the password is entered, it is transmitted across the network in encrypted form, so eavesdropping on the data transmission will not reveal the password. Subsequent transmissions of the password to "login" onto the device are encrypted and "safe". The only case when the password is transmitted across the network "in the open", is when it is being changed and submitted in **General Setting** form. Therefore, you must set passwords in the secure environment where you can make sure that no one is "eavesdropping".

SNMP communication
SNMPv1 does not implement encryption. Authentication of clients is performed only by a "community string", which is transmitted in clear text. SNMP communication should be used in trusted networks and disabled if not used.

XML/JSON operation
A challenge-response authentication can be used in login process. The password can be transmitted by custom application across the network in encrypted form.

Web and XML/JSON access can be restricted by IP Address (range of IP Addresses) or by MAC Address.

11. I/O Ports

11.1. Digital Outputs Ports JP1/JP3

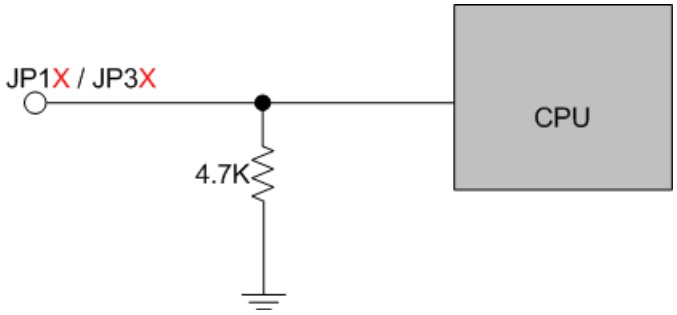

Figure 44. DAEnetIP4 Digital outputs port JP1/JP3

- Maximum output current sink by any Digital Output of JP1/JP3 - **2mA**;
- Maximum output voltage at low level is **0.4V DC**;
- Minimum output voltage at high level is **2.4V DC**;
- There is pull-down resistor of 4.7K connected to GND inside **DAEnetIP4** for every JP1/JP3 output pin.

The Digital Outputs ports are not protected and they are connected directly to the CPU! Over-current will damage the CPU!

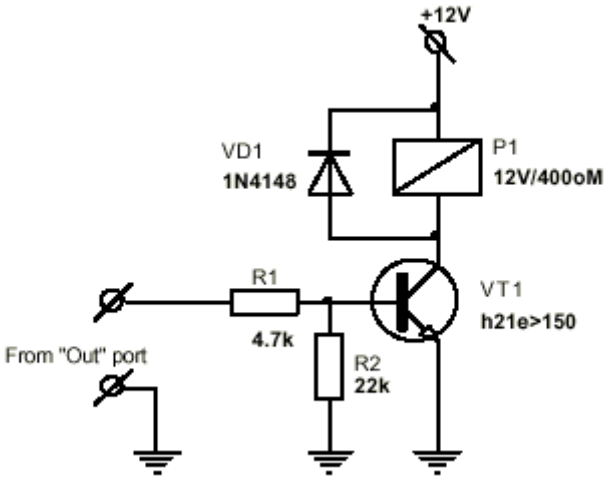

Figure 45. Example how to connect SPDT relay to Digital Output

11.2. Digital Inputs Port JP2

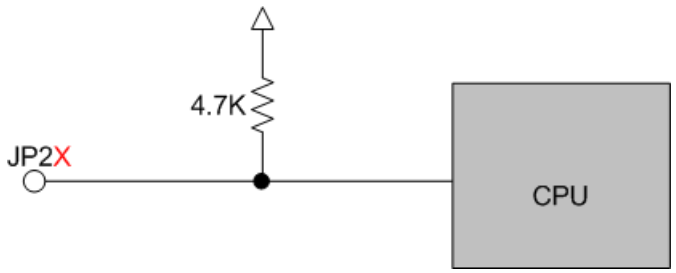

Figure 46. DAEnetIP4 Digital Inputs port JP2

- Low level (0) input voltage: **From 0V DC up to 0.66V DC**;
- High level (1) input voltage: From **1.63V DC up to 5.5V DC**;
- Maximum input voltage on any JP2 input pin is **5.5V DC**;
- There is pull-up resistor connected to 3.3V inside **DAEnetIP4** for every JP2 input pin.

The digital Inputs port is not protected and it is connected directly to the CPU! Over-voltage or inverse polarity voltage will damage the CPU.

It is recommended always to use resistor in sequence to the Digital Input.

It is recommended to use protection resistor with a value of about 1K in order to connect the JP2 Digital Inputs port pin with GND via switch or button. Bellow is shown an example how to connect button to **DAEnetIP4** JP2 port.

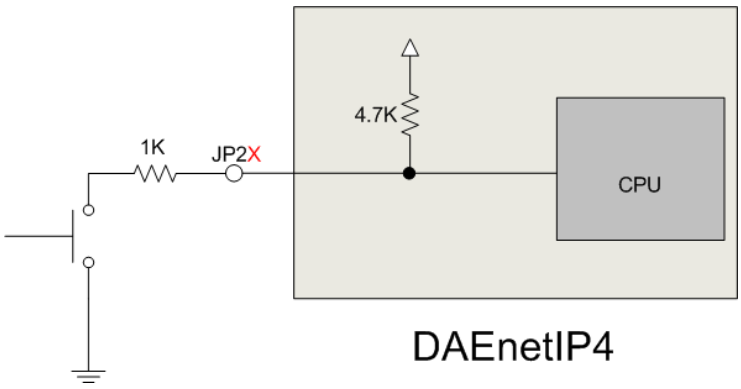

Figure 47. How to connect button to Digital Inputs port JP2

11.3. Analog Inputs Port JP4

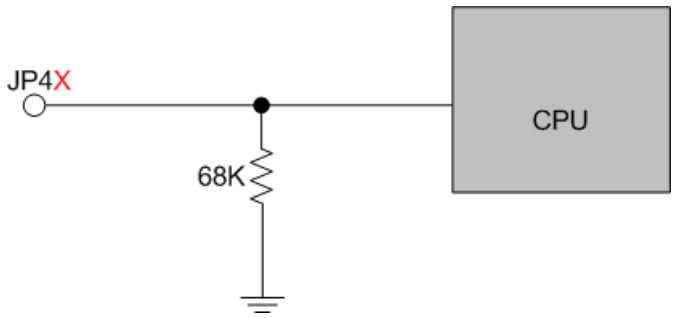

Figure 48. DAEnetIP4 Analog Inputs port JP4

- There is pull-down resistor of 68K connected to GND inside **DAEnetIP4** for every JP4 input pin;
- Maximum input voltage on any JP4 input pin should not be more than **2.048V DC;**

The Analog Inputs port is not protected and it is connected directly to the CPU! Over-voltage or inverse polarity voltage will damage the CPU.

It is recommended to use always resistor in sequence to the Analog Input.

Bellow is shown example how to extend the JP4 Analog Inputs port to measure voltage 0-10V DC. The 270K resistor also acts as protection.

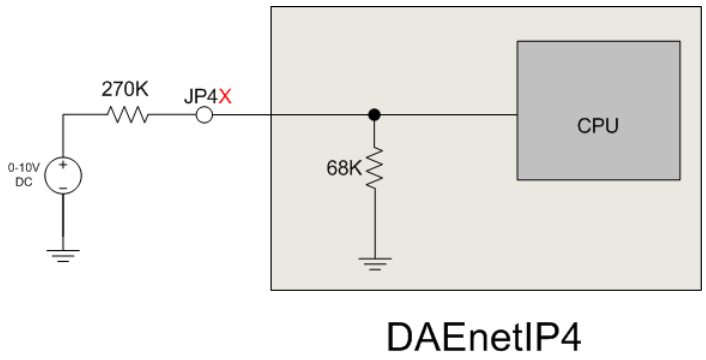

Figure 49. How to extend **DAEnetIP4** Analog Inputs port JP4 to 0-10V

11.4. PWM Port JP6

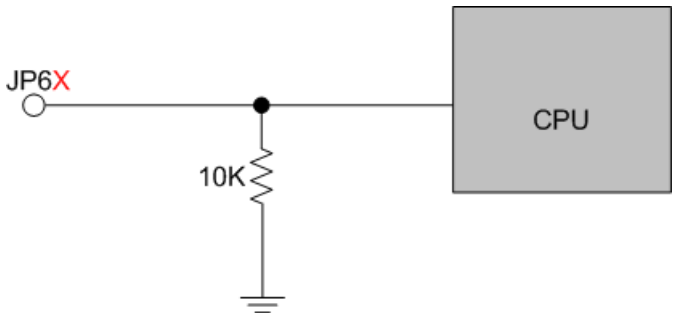

Figure 50. DAEnetIP4 PWM port JP6

- There is pull-down resistor of 10K connected to GND inside **DAEnetIP4** for every JP6 input pin;

The PWM port is not protected and it is connected directly to the CPU! Over-current will damage the CPU!

In order to be used, the PWM port should be connected to additional logic like MOSFET-s or other drivers to ensure proper controlling power for the load.

12. PCB mechanical drawing

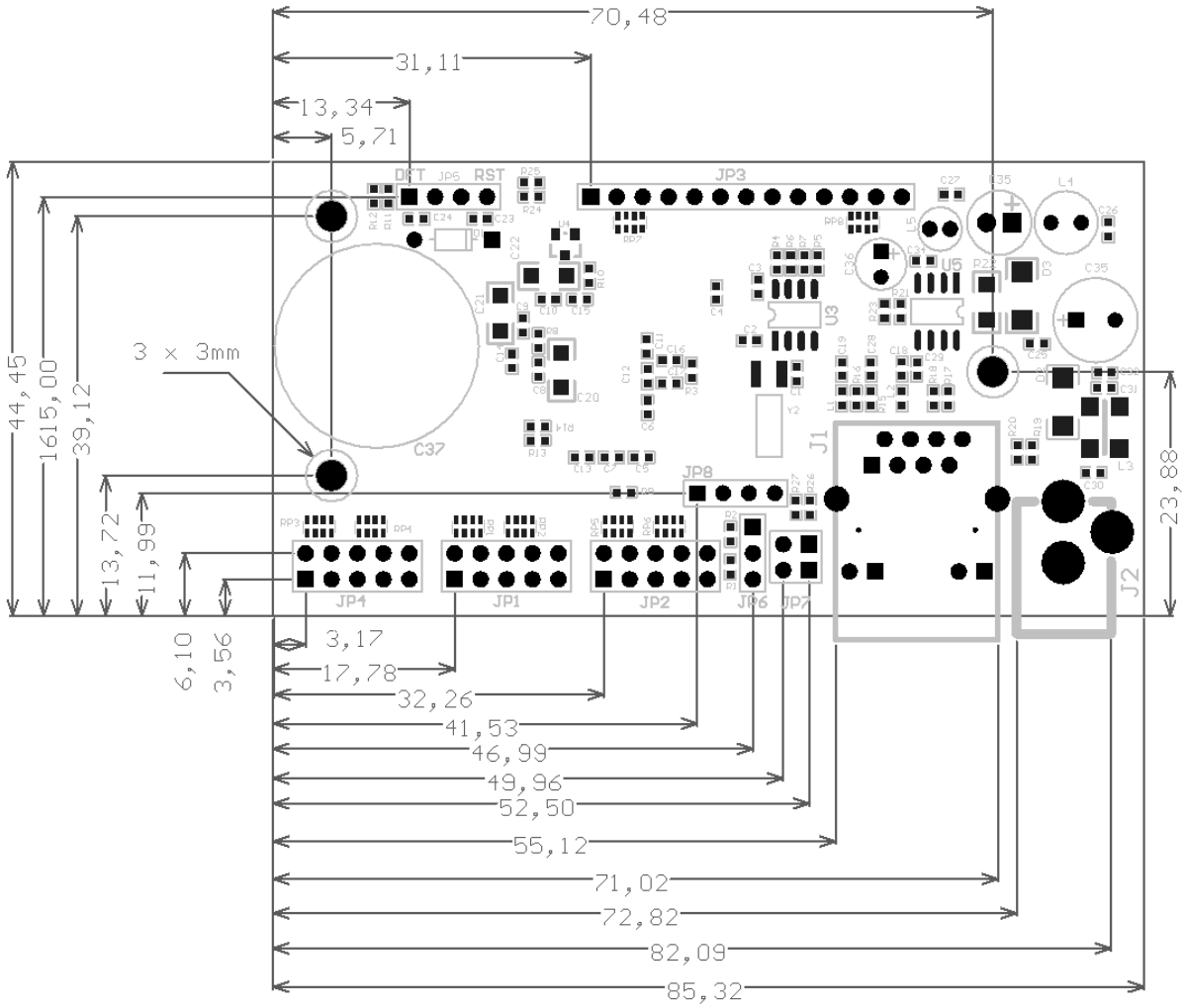

Figure 51. PCB drawing (top view)