NetControl

User's

Manual

rev. 2.35 (reflects firmware v5.62)

20.05.2024

TABLE OF CONTENTS

©2008-2024 www.ipnetcontrol.net

Legend:

The text contains additional and useful information that explains specific situations and features.

The text contains information of essential importance which you must get to know well!

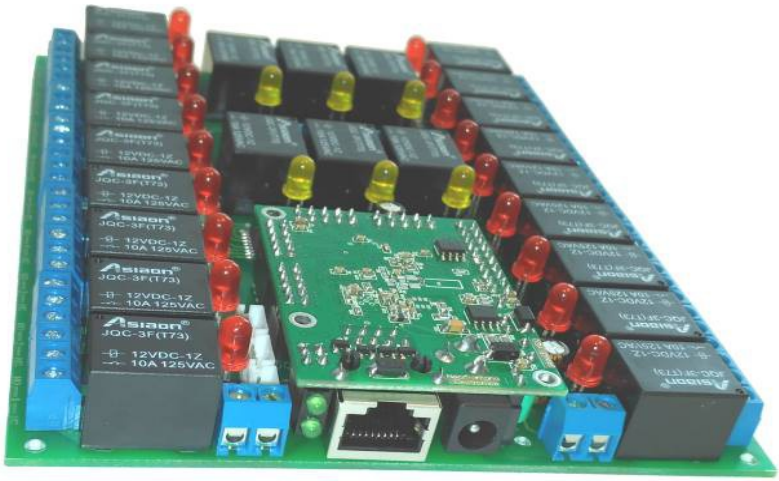
©2008-2024 www.ipnetcontrol.net

Version	Date	Short description of the made changes
2.35	20.05.2024	Changed description of time period accuracy (3.3.1, 3.4) from v5.62 Changed the structure of the document - information about individual models is separated in a separate document. Added WiFi module section.
2.32	03.11.2023	Added descriptionc for new steps in Macros, Circular Event Log, 'Virtual IO' and outputs filter of ON commands (v5.58 and 5.59)
2.31	03.08.2023	Added chapter for ModbusTCP access, added in v5.56 Added chapter for new function 'Services access list" from v5.56
2.30	12.06.2023	Added description for „Events log" function, introduced in v5.55
2.29	10.03.2023	Description added for models 8CT1RS и 7R8A1RS
2.28	12.10.2022	Added second note for "Remote IO Action" section
2.27	09.06.2022	A lot of changes in IO Settings and Automation sections, concerning latest firmware versions
2.26	09.11.2021	Corrections in 3.3.2(MA filter), in 3.2.1- HTTP Disable, in 4.1.1 - added ioValueFiltered and new section 3.8.5.
2.25	10.09.2021	Added section 3.2.4 for the list of host names (from v5.37). Added new texts in the DHCP section.
2.24	05.07.2021	Rework on sections 3.2.1 and 3.8.1 for second Web user „IO User" (from v5.35)
2.23	18.06.2021	Added subsection 3.8.6 for configuration save/restore from file (firmware v5.34)
2.22	29.01.2021	Added description of two new models for DIN rail mounting: 4R3OC7A and 4R6I2O Description of 'Macros' is modified – added step „Stop Macros" (firmware v5.31)
2.21	28.07.2020	Added description for new function "Remote IO Action" in Macros, introduced in firmware v5.22
2.20	17.06.2020	Added description for new function "Timers" introduced in firmware v5.21
2.19	22.05.2020	A description (3.3.3) has been added to define up to 2 pcs. consumer sensors; added information for macro startup via MQTT (firmware v5.19)
2.18	30.04.2020	Added description for new model 6R8A.
2.17	07.02.2020	Added description for new model 2R4A.
2.16	02.12.2019	Added description for new MQTT option "Mirror /in to /out" (available in v5.15)
2.15	23.10.2019	Total re-work for changes in firmware v5.12. Added 4R4S1A WiFi model
2.13	18.01.2019	Sections 1.5 and 2.1.1 added (for HTTP URL commands)
2.12	11.06.2018	Added description for MQTT, available as of version 5.7 Appendix added with description of channel types
2.11	05.02.2018	New model 24R3S2A is added. Added a note about the 'Last' option of 'Default' parameter.
2.10	27.11.2017	Corrections in text concerning all new models with increased number of inputs for external sensors.
2.9	16.11.2017	Descriptions added for new function in firmware v3.17: HTTP auth brute-force blocking, ACT LED mode, Ping data size
2.8	03.05.2017	Added description for new SNMPobject ioPulseCfg[P] (firmware v.3.16 or newer)
2.7	07.10.2016	Changed description of '>HIGH', '<LOW' and '>HIGH or <LOW' modes in section 2.5; new models added.
2.6	09.06.2016	Notes are added about 4RU1SH2S and wrong 'Cloud Scale Coefficient'
2.5	11.05.2015	Added section for 4RU1SH2S model
2.4	19.03.2015	Correction primary concerning new model 4PC2R and firmware 3.9; ioGauge object added. Added description for 4PH1R
2.3	02.2015	Added description for calculations required for use of hummidity sensor HDS300
2.2	07.2014	Description for the new option for digital outputs 'Invert Output' Added description for DHCP mode of configuration and 'swap-server' option available in version 3.3
2.1	07.2014	New model ***NetControl 8R1T1A***
2.0	05.2014	Initial version of the document for software 3.xx

©2008-2024 www.ipnetcontrol.net

1. Introduction

NetControl is a network device with 10/100Mbit Ethernet interface, its core is built with the popular ***PicoIP/PicoIPv2*** module, developed by NEOMAONTANA ELECTRONICS. In order to implement optimally all the features of the ***NetControl*** series, it has been developed a special software version which has been unified for all the models of the device.



Depending on the model different input output circuits are incorporated in the device: relay outputs, inputs for temperature sensors, alarm inputs for contact sensors, power supply module with wide range of the input.

As a network device ***NetControl*** supports the following protocols and functions:

- The compulsory network protocols ARP, IP, ICMP (ping), DHCP, TCP/IP
- 802.1q VLAN support with a possibility to work in the full 12bit VLAN range
- SNMPv1 protocol for access to all parameters and functions of the module
- MQTT protocol for automation and IoT solutions (needs v5.7+)
- SNMP-Trap messages generation with inputs change
- ModbusTCP access to IO
- Output signals generation on/with analog inputs change
- Web Server to access to all parameters and functions of the module
- Authorization mode for Web access
- Access prohibition possibility on SNMP for configuration
- Access prohibition possibility on SNMP/Web via network (IP/Mask)
- Possibility for disabling Web access
- TFTP client for update of the system software
- Access to the device through cloud platform of **domo.ipnetcontrol.net**
- 8(24)-channel 'ping' monitor functionality
- SNTP client for time synchronization of internal clock for timers execution
- and much more functions, also by customer request

With the help of ***NetControl*** the following problems are successfully solved:

- Active monitoring and control of network segments and routes
- IP security of telecommunication cabinets and other objects
- Temperature monitoring
- Measuring the analog values- voltages, electric currents, etc. (depending on the model)
- Smart Home / Home Automation
- Wireless control of heating and other systems via Internet
- And many more…

©2008-2024 www.ipnetcontrol.net

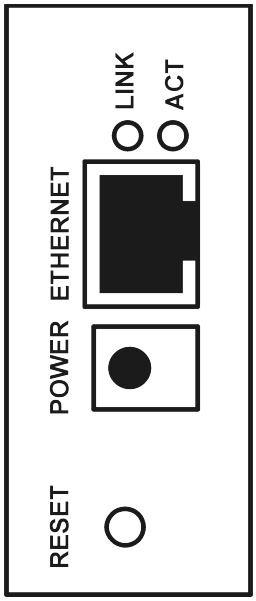
1.1. Main panel (for all models in gray ABS box). Restoring factory settings

All devices in the ***NetControl*** series have 10/100Mbit Ethernet (10M for firmware 3.x) interface RJ-45 connector located on the main panel. Establishing a connection with the other device is indicated by the LED on the main panel 'Link'. When there is input/output traffic the LED flashes.

The factory default IP settings are:

IP address: 192.168.1.100
NetMask: 255.255.255.0
Gateway: 192.168.1.1

Before you try to access the device make sure that you have regular network access to its IP address. For this purpose you can use the command 'ping 192.168.1.100'. In case you do not receive a response from the device , you may have a problem with the network settings of the device, of the network which you are using it or with the computer from which you are connecting.



POWER	Power port 5.5x2.1 12VDC +-10%. The consumed power depends on the model.
ETHERNET	RJ45-F, 10/100Mbit AutoMDIX (10Mb, no cross-over autodetect for v3.xx) with differential over-voltage protection (10V)
RESET	Resetting the device to its default factory setting happens after you hold the RESET button while powering the device. The LED „ACT" starts flashing when settings are reset, then you can release the button.
LINK Led	It glows when Ethernet connection is established. It is flashing when there is a traffic flow
ACT Led	By default it is always glowing when it's powered. Other functions could be set in the software.

This box has dimensions 118x72x35mm. At the bottom of the device there are provided holes for wall mounting via of two screws. The distance between the screws should be 70mm.

©2008-2024 www.ipnetcontrol.net

1.2. DIN rail enclosure

Some models are designed in DIN rail compatible enclosure with 105mm width (105x65x90mm) and ABS V0 (UL94V-0) material.

Depending on the model there are different screw terminal configuration, but on bottom row is always accessible the RJ45 connector and RESET button near.

©2008-2024 www.ipnetcontrol.net

2. Access to the device

2.1. Access via the cloud platform of domo.ipnetcontrol.net

The “SmartSpace Cloud” is server-based platform for communication with ***NetControl*** devices. It has modern Web interface accessible 24/7 from your computer, tablet or smart phone. Special “lite” version is also available to for easier access to primary functions via smart phohe.

NetControl devices connect to platform automatically (when SPC mode is enabaled) without the need for any specific network settings except standard network parameters (IP/MASK/GATEWAY) and Internet connection supplied by the gateway.

Inside the cloud you have unlimited and constant access to:

1. All available input/output circuits of ***NetControl***-relay outputs, voltage, temperature ,humidity, energy measurement, etc., alarm input and more.
2. History and reports for all data from and to the device.
3. Energy consumption and callculations
4. Alerting (by e-mail, optionally SMS) by user’s set events
5. Powerful module for creation of automatic tasks/scenarios (Macros): here you define the conditions (combination of time and/or information from your devices) at the beginning of which the series of actions that you have set will begin. This series of actions itself could contain conditions for the state of the devices or time according to which the actions set out in a manner that you chose.

All you need to ‘get’ in the cloud is:

1. ***NetControl*** device (version 3.00 or newer)
2. Internet access for the device
2. Free registration at https://domo.ipnetcontrol.net

2.2. Access via Web browser

NetControl has an integrated HTTP server that corresponds to the standard TCP 80 port.

To access the device just type its address in the browser (Firefox, IE, Chrome…), for example http://192.168.1.100. You will be asked to type username and password. Their factory settings are:

User=admin, Password=admin.

The WEB server has a special mode ("HTTP auth brute-force blocking") for blocking of brute-force attacks to its user/password. After 10 consecutive failed attempts for login – the WEB access is totally blocked ('503 Service Unavailable' page is displayed) for next 5 minutes. After this time – access to WEB server is granted again. IP address of last successful login is not blocked by this mechanism. This function is enabled by default and as of v3.17 it can be enabled/disabled by user (from "IP Settings“ menu).

2.3. Access via HTTP URL commands

For integration into automated systems, scripts, etc. it is often necessary to directly send commands to the device via HTTP. You can find information about the data format in section 3.1.1.

2.4. Access via SNMP protocol

NetControl supports the main commands for SNMPv1: *snmpget* and *snmpset*. With their help the values of the configuration parameters can be read or changed. They are

described in detail in a special MIB file that is available for downloading on http://www.ipnetcontrol.net.

Two passwords are used for access via SNMP (community strings): Read-Only Community String and Read-Write Community String. With the first one only reading of the parameters is possible and with the second they could be alsochanged.

The default passwords are: **Read-Only=public, Read-Write=private**

For more information and examples for using SNMP commands go to page 48.

When using SNMP for access, use snmpget and snmpset only toward one OID rather than to a group of OIDs. Other commands (for example snmpwalk) are not supported!

2.5. Access via MQTT

From version 5.7 of the system software (PicoIPv2 core only), NetControl's I / O circuits are available for access and control over MQTT - a popular and lightweight communication protocol for centralized management and monitoring of IoT network devices.

MQTT communication is implemented as an alternative to the SmartSpace Cloud connection and the user has to choose which of the two modes of control to use (or completely disable the external control).

For more detailed information, please refer to chapter 5.

2.6. Access via ModbusTCP (from v5.56)

Since this release, access to the NetControl I/O circuits has been added via the ModbusTCP protocol. The protocol is common in PLC systems, solar inverters and other automation systems.

The embedded ModbusTCP server is listening on standard port 502 (TCP).

Since the protocol has no built-in means of authorization and security, it is recommended to combine its use with the function of restricting access to server services by IP address (see 3.2.5). By default, ModbusTCP is disabled in the IP filter factory settings.

The following table lists all supported access addresses, as well as the type of commands with which they can be accessed.

Server supports only ONE active TCP session for ModbusTCP access!

Access Function (FuncCode)	Address	Description
ReadCoils (1) WriteSingleCoil (5) WriteMultipleCoils (15)	0 ... 23	For 1bit access to the digital input-output channels (ON=1, OFF=0)
ReadHoldingRegisters (03) ReadInputRegister (04) WriteSingleRegister (06) WriteMultipleRegisters (16)	0 … 32	16bit access to all I/O circuits. For digital ON=0x0001, OFF=0x0000; for analog (sensors) – 10bit value (0...1023) of the ADC (read only).
ReadInputRegister (04)	1000 ... 1063 1064 … 1079 (v5.59)	32bit (BigEndian) access (registers are in pairs, eg 1000-1001, 1062-1063 for every IO channel) to all 32 input-output circuits. For digital, the value is again 0/1 for OFF/ON. For analog/sensor - the value is 32bit float, which corresponds to the value of the set sensor for the input (eg for TDS300 the value will be 23.5 degrees). For pulse counter type inputs – the value of the counter (it is 32bit). For Vritual IO (from v5.59), the range 1064...1079 is used (in practice, virtual ports appear after the 32 hardware ones).
ReadDeviceIdentification (43)		The response to this request contains only the data from the Basic category: (0VendorName=“Neomontana Electronics” (1)ProductCode=”NetControl MOD=xx” (2)MajorMinorRevision=”V5.56” / 'xx' is code for the specific NetControl model /

Attempting to read addresses outside of those specified will result in an "IllegalAddress" error message being returned. Write commands can be issued to analog input addresses - not treated as an error, but nothing will be executed.

Registers in the range 1000 ...1062 must always be read in pairs (or an even total) and the starting address must also be even! Otherwise, an "IllegalAddress" error message will be returned.

The 32 (23) access addresses to the I/O circuits correspond to the SNMP "Number [P]" of the channels available for each model (see the description of the specific model in its "Relationship between SNMP index and channel type" section). The only difference is that in SNMP the first channel is numbered 1 and in ModbusTCP – 0 (you have to subtract 1 from 'Number [P]' to get the address in ModbusTCP). It is acceptable (for unification) to always read all addresses, but those that have no real I/O for a model will return zero.

©2008-2024 www.ipnetcontrol.net

3. Control/configuration of the device through the Web

3.1. Control/status of the input/output channels (menu ‘Status’)

This is how it looks the initial Web page loaded after successful authorization (according to the model there could be seen different number of inputs/outputs). You can also reach it via ‘Status’ menu.

Selecting the ‘My NetControl’ name off the device you can enter the settings section (where the listed below menus are located).

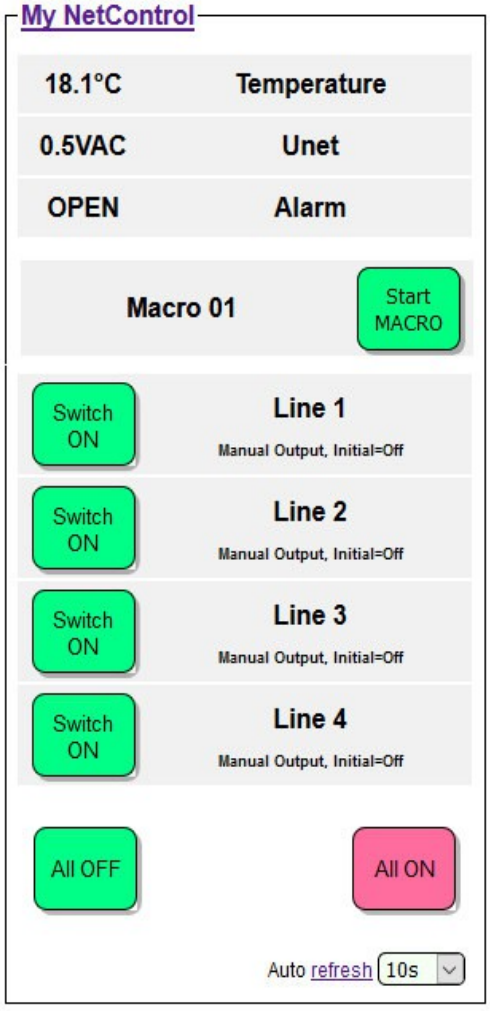
At the top of the page are the analog inputs (temperature, voltage, etc.) with their names and current value.

Following are the macros, defined by user I set up to be accessible on this main page. Displayed macros can be started with a button click.

The next group is the digital circuits which could be inputs or outputs according to the device. Through the buttons you can change their condition (if they are output) as well as their current condition can be seen. Under the name of the channel is described the chosen mode of work.

With the ‘All On/All OFF’ buttons you can turn on or off all relay outputs with just one click (this command doesn't influence macros).

You can set a time for an automatic refreshment of this page if you want to automatically monitor the values of the channel.



3.1.1 Control / status of I / O through URL commands

If you need to submit a command via HTTP URL to turn on or off an output (for example, through a script), you must use the HTTP GET request at the following address:

http: // <NetControl IP address> /inchange.cgi?ref=ioreg.js&PP=VV

, where

- **PP** is ***NetControl's*** channel number of (in double-digit HEX number with leading zero). For a specific model, channel numbers can be seen in the description of SNMP objects, **using the value of [P] of SNMP minus 1**. For example, NetControl 8Rxxxx Line1 has PP = 08, Line2 PP = 09, Line3 PP = 0A ... Line8 PP = 0F;

You can use PP = FF - then the command given will be applied to ALL output channels (relays). This is equivalent to the AllOn / AllOFF commands from the Web interface

- **VV** is the value to be fed to the output: values '00' = OFF and '01' = ON.

Example with 'curl' (switch on relay 9, ie Line 2):

>curl -u admin:admin "http: // <NetControl IP address> /iochange.cgi?ref=ioreg.js&09=01"

NetControl will respond with the 'ioreg.js' JavaScript file (if you are not interested in file data, you can replace 'ioreg.js' with 're-done' in the query and then return a small HTML file). The JS file contains the 'IO' variable which represents the status of all I / O channels as a comma-separated string with hexadecimal values:

IO = "00,00,00,00,00,00,00,00,01,00,00,01,00,00,00,00,00,01,01,00,00,00,00,00 , 01CB, 0015,000C, 005A, 005A, 0068,03FD, 0145 "

The first number refers to a channel with [P] = 1, the second for [P] = 2 .... and so for all 24 digital input / output circuits (not all of them are available in the particular NetControl model). Here, the value 00 corresponds to the off, and 01 to the on state.

The last 8 numbers are the values for the 8 analogue-to-digital converter inputs (0 to 1023), and the connection to the specific ***NetControl*** sensors is again through the channel number [P] of the SNMP access objects.

All numbers in IO are in hexadecimal format!

By reading directly to "http: // <NetControl IP address> /ioreg.js", you can always access the current state of the inputs and outputs.

There is a demonstration PHP script (available for download at www.ipnetcontrol.net) how to access ***NetControl*** with HTTP URLs or SNMPv1.

3.2. Network parameters and settings(menu 'IP settings')

IMPORTANT!!! Due to the specifics of the SNMP protocol that is supported by the device (inability to simultaneously access multiple OID), it is desirable the initial set up of IP/Mask/Gateway to be done via Web. Otherwise you may be not able to set the desired settings because of the restriction that they should be changed individually.

3.2.1 'IP Configuration' section

Software Version and Uptime	Current software version of the device and its working time since last restart.
MAC address	MAC address of the device (it should be unique within the range of the local network)
Ethernet Settings	Link to a page, containing settings for the Ethernet link. See section after table for more information.
WiFi enable mode	WiFi Module Mode (Only available with Models with Extra WiFi Module). Two modes are available: **'When Ethernet Down':** WiFi is activated when the Ethernet connection fails (ie, physically removing the cable or stopping the opposite device). When Ethernet link is again established – WiFi automatically stops. **'Never':** WiFi module is disabled
IP address, Subnet mask, Default gateway	Standard network settings for the device, mask and access gateway from/to Internet/external networks
DHCP client	Determines whether the parameters-address, mask, gateway are set statically or are dynamically allocated via DHCP protocol
Tagged VLAN mode	Activate tagged VLAN mode (802.1q)
VLAN ID	Tag (ID) of VLAN in mode „Tagged VLAN mode=Enable"
Access MAC address 1 and 2	Limit the access to the device up to two MAC addresses. Zero value of an address is not taken under consideration (accordingly

/ Global access filters/	too two zero values = no access limitation). *If you are using protection according to MAC address keep in mind that when there is access by external networks towards the module are arriving packets with the MAC address of the Default Gateway. In that case it must be always set as one of the two addresses with access.*
Network IP/MASK / Global access filters/	Locking the access to the device only from the set network/mask. Mask 0.0.0.0=no locking.Outgoing services (like SPC, MQTT) are not affected by this filter. *Protection according to MAC address is with a HIGHER priority than the one according to IP/MASK!* *If you must change the network parameters for access through SNMP first set the IP address at open mask (0.0.0.0), and after that the mask. B otherwise (at changing first the IP address, at some set mask) it could appear an unwanted combination IP/MASK and the access will be blocked.*
External managing service	From here, you can specify which protocol to control the device remotely: SmartSpaceCloud (domo.ipnetcontrol.net), MQTT or without external control. Below this setting is shown the status of selected service: ONLINE / OFFLINE.
TFTP client	Enable/Disable updating via TFTP
TFTP server IP address	IP address of the server that contains the updating image. For more information about updating via TFTP go to page 32.
SNTP server IP address	IP address of NTP/SNTP Time server for clock synchronization (required for operation of the Timers module). Value 0.0.0.0 - disables synchronization and the Timers module.
HTTP access (from v5.38)	Disable Web access. It is intended to use for security purposes, when a module that uses other communication protocols (SNMP, MQTT, SPC) is already configured and installed.
HTTP listening port	Allows you to set the TCP port number on which to 'listen' the Web server. Permitted values are 1024 ... 65535. Default is 80.
HTTP auth brute-force blocking (from v3.17)	This option allows user to activate/deactivate this function. More information on the function itself can be found in 2.2.
ACT LED mode (from v3.17)	The ACT Led behavior is defined here: -"Power ON": the led is powered on all the time -"Incoming PING request": led flashes on every received „ICMP Echo request" towards its IP address -"Incoming PING reply": led flashes on every received „ICMP Echo reply". Typically this is a response from one of the hosts, configured in "PNIG Monitor". -"Incoming PING Any": flashes in both two previous cases -"DHCP Valid IP": When in DHCP mode, led is on when valid IP is retreived from DHCP server. In static IP mode the led mode is equivalent to "Power ON'.
IO User web account	Settings for the second (non-administrative) Web access user - "IO User". It can be activated or deactivated, as well as assigned a

(from v5.35)	mode of access to the title page: Read Only or Action. *By default (or after an update to an older version) the second user is not defined and in order to function it is necessary after setting in Enabled mode to set a name and password from the menu "Misc". Changing user to "Disabled" does not delete the username/password. When reactivated with "Enabled", the last username and password will be valid.*

„Ethernet Settings"

From this submenu you can set the physical parameters and Ethernet connection mode. You can set the manual speed and duplex mode and disable the automatic recognition of RX / TX twisted pairs. You can access information in which mode the current connection is established.

In the general case (Link = Auto, Auto MDIX = Enabled) no changes need to be made unless there are problems with the connection to the opposite device.

„WiFi Settings"

This menu is only available if there is a WiFi module. For more detailed information refer section Error: Reference source not found.

3.2.2 'SNMPv1 access settings' и 'SNMPv1 traps/remote IO settings' sections

This section contains all the necessary settings for accessing the device via SNMP protocol.

Listen on UDP port	Determines the UDP port on which the SNMP server 'listens' in the device. Valid values 161 (standard) or 1024...65535.
SNMP access to IP configuration	Determines whether it is possible via SNMP to change some parameters of the IP settings of the device. Does not affect commands for reading/recording from input/outputs ports
SNMP read-only community string	Read only password (4-13 symbols in Latin characters)
SNMP read-write community string	Read and write password (4-13 symbols in Latin characters)

In the next section are the settings related to the generation of SNMP automatic "trap" messages from ***NetControl***. To use this functionality, a special server is needed to process these messages.

The setting is combined with the parameters for a second ***NetControl*** device to be controlled via Macros. For more information see section 3.4.1.

Target IP	IP address of the SNMP trap server to which the generated SNMP traps are to be sent. Messages are sent to standard port 162 (UDP). This IP address is also used for "Remote IO" - sending commands (via Macros) to another ***NetControl*** device.
Community string	Password for the TRAP server (4-13 characters in Latin) and at the same time for "Remote IO".
UDP port for remote IO	Specifies the UDP port to which SNMP "Remote IO" messages will be sent. By default, SNMP uses port 161 (UDP), but other values can be set.

©2008-2024 www.ipnetcontrol.net

3.2.3 DHCP - dynamic distributing of the general network parameters

The device supports DHCP protocol which allows the following parameters (if there is a DHCP server in the network) to be dynamically distributed:

-IP address
-Network mask
-Default Gateway
-DNS servers (***option domain-name-servers***) (2pcs) (from v5.37)
-TFTP (***option swap-server***) server for updating (after v3.3)
-NTP/SNTP (***option ntp-servers***) time server

The first three parameters - IP/Mask/Gateway are the required minimum for the device to function properly. The other parameters, if not passed via the DHCP protocol, use the last set of static values from the settings.

Through DHCP, you can also choose from hostnames list (only for TFTP and SNTP), although DHCP does not directly allow this. To do this, you must set **0.0.1.D** as the IP address of the option, where D = [1..10] corresponds to the host name number from the "DNS names cache" list (see 3.2.4).

__NetControl__ is storing all of its settings in a non-volatile (FLASH) memory including the network ones. The parameters that are distributed through the DHCP do NOT lead to updating these in the FLASH memory (unless the Web page with the settings is not used while in DHCP mode and „Apply Settings" is clicked - then the current values of the DHCP will be stored in the memory). Because of that when the DHCP option is being turned off the device adopt the last recorded settings in the memory.

Enabling the DHCP client in the absence of a working server (or existing network problem) can prevent __NetControl__ to load its network parameters and thus the access to it to be lost. To prevent this __NetControl__ waits for a certain period of time (about 40s after its restart) to receive its network settings. In case that does not happen, __NetControl__ loads the most recently set static parameters and starts working with them while continuing to search the DHCP server. When the response of the server is received __NetControl__ immediately adopts its new dynamic parameters.

When the access via Web and enabled DHCP mode in the column of the relevant parameters is shown the data obtained from the DHCP server, not the statically set parameters! If you want to see the currently set static settings turn off the option (without confirming with the button „Apply Settings") and they will automatically appear in the fields.

©2008-2024 www.ipnetcontrol.net

File: dhcpd.conf (partitial content)

```
#
# Sample configuration file for ISC dhcpd for Debian
#
# $Id: dhcpd.conf,v 1.4.2.2 2002/07/10 03:50:33 peloy Exp $
#
# option definitions common to all supported networks...

option subnet-mask 255.255.255.0;
default-lease-time 600;
max-lease-time 7200;

subnet 192.168.1.0 netmask 255.255.255.0 {
        range 192.168.1.160 192.168.1.175;
        option domain-name-servers 192.168.1.1;
        option routers 192.168.1.1;
        option subnet-mask 255.255.255.0;
        default-lease-time 3600;
        max-lease-time 7200;
}

host pico_test {
  hardware ethernet ec:f2:36:00:0b:db;
  fixed-address 192.168.1.158;
  option swap-server 192.168.1.104;
  option ntp-servers 192.168.1.1;
  #Use the second  host name from user cache as SNTP server (e.g. 0.bg.pool.ntp.org)
  #option ntp-servers 0.0.1.2;
}
```

The example briefly illustrates the settings in dhcpd.conf (Linux) for dynamic configuration in network 192.168.1.0. On a particular device ('pico_test', distinguished by its MAC address) they are individually fixed IP/Mask/Gateway, TFTP and SNTP.

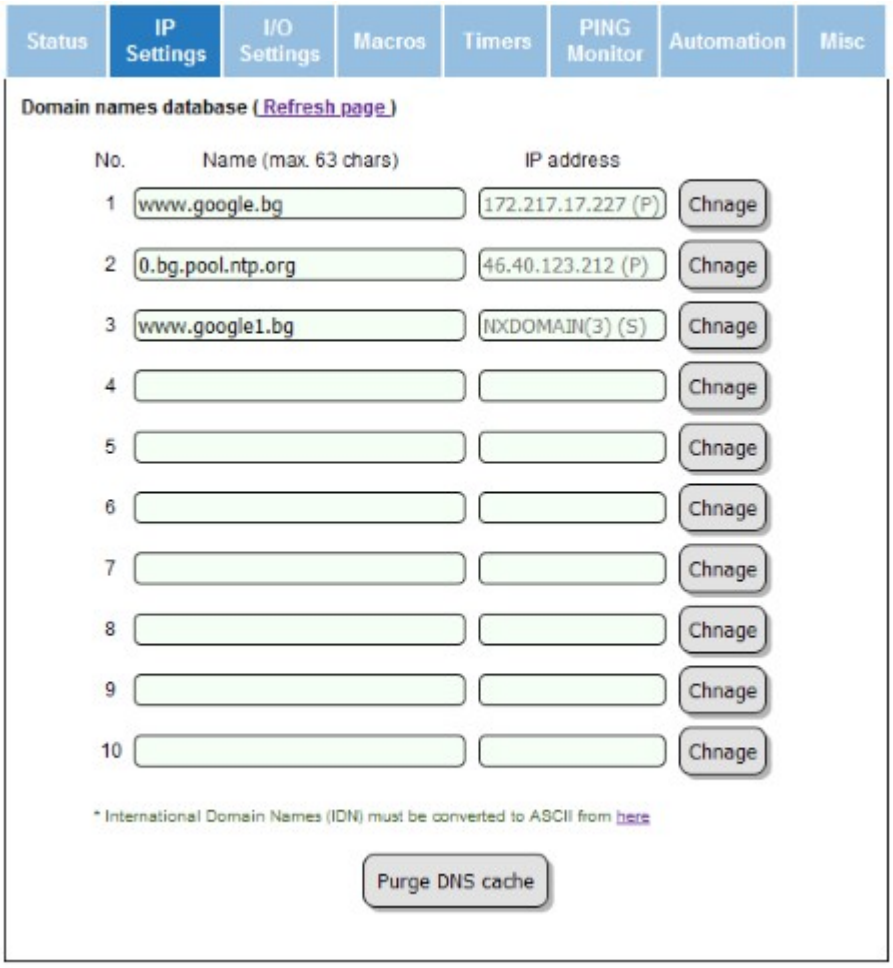
3.2.4 User-defined hostnames list, that can be used by different services

As of v5.37, a list of 10 hostnames has been added to the system software, which can be selected as addresses for the TFTP, SNTP, SNMP, MQTT and Ping Monitor services. For each service, the user chooses whether to use a DNS list name or an IP address. The list can be accessed from the link "Manage DNS names cache" in the menu "IP Settings".

A field for entering the host name is available for each of the 10 records (e.g. 'yahoo.com'); each newly introduced host must be confirmed with the corresponding "Change" button.

The "IP address" field will display the IP address returned by the DNS servers or an error message with its RCODE. The most common errors you will see are **NXDOMAIN** - such a domain/name does not exist or **NOIP** - the name exists, but has no IP address. You will also see (P) or (S) symbols - this will show you which of the DNS servers (Primary/Secondary) returned the corresponding result.

©2008-2024 www.ipnetcontrol.net



The "Pruge DNS cache" button allows you to clear the available IP address information and query the DNS servers again. If you want to use IDN names (e.g. containing Cyrillic and other characters), you must first convert them to ASCII (via the link provided on the page).

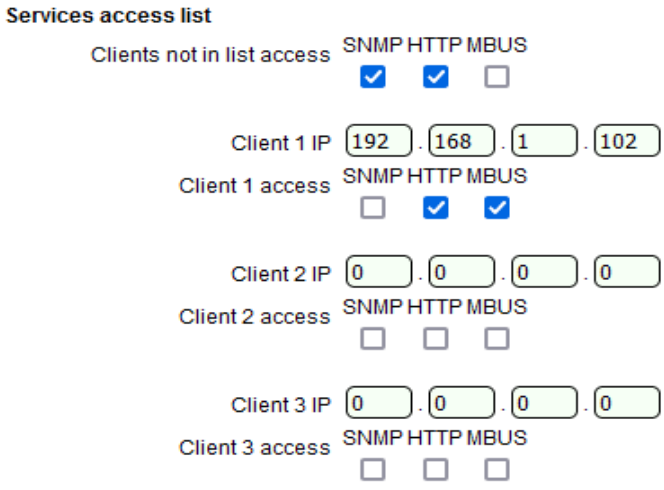
The use of hostnames is more convenient, but it should be borne in mind that the process of converting the name to IP (which actually works in the TCP / IP stack) adds additional latency to the processes. With Ping Monitor, this could lead to an incorrect timeout.

3.2.5 Access IP filter for server services (SNMP, HTTP, ModBUS)

In version 5.56, the basic functionality (“Global access filters”) for limiting access to the services provided has been expanded: "IP settings"→"Services access list".

This new filter allows the description of up to 3 IP addresses, with individual settings for which services are allowed or denied access. Services are only those that are in “listen” mode, for example SNMP, HTTP (respectively, these filters have no impact on outgoing services such as MQTT, SPC, etc.).

In addition to the individual settings for the services of each filter, there is also a global group of settings ("Clients not in list access"), which is valid for



©2008-2024 www.ipnetcontrol.net

those IP addresses that did not find a match in the list. Setting the IP address to 0.0.0.0 effectively disables the line from the list.

In the given example for the IP address 192.168.1.102, only HTTP and MBUS access is allowed, and SNMP access is prohibited. For any other client IP addresses, SNMP and HTTP are allowed, but MBUS is prohibited.

3.3. Input/Output channels. Operation mode and parameters. (menu 'IO Settings')

In this menu are gathered all settings concerning the existing input/output circuits in the device: relays, temperature inputs, voltage, etc.

At the top of the page you can write a name for your device '**Host name**'. From the information about the model which is also written next to the name of the device you will be directed after clicking straightly to the description of the particular model.

You can also put your names on every one of the existing channels. The name will be visible on the home page as well as it can be obtained via SNMP request.

The confirmation of the names is done by a separate button: 'Change Names'. If you first confirm other parameters with the button 'Change Parameters' before you select 'Change Names' then the names will not be saved. Both buttons work independently from each other!!!

3.3.1 Digital I/O Channels

Depending on the model ***NetControl*** has different digital input-output circuits – relays, digital inputs and more. All are available for configuration in this section.

Visible	Determines whether the circuit to be visible on the home page. It does not affect the access through SNMP; it does not suspend the All ON/All Off commands.
Name	Name of the channel (change of the names is confirmed with the „Change Names" button)
Default	Sets the initial state of the input/output after the device has been powered up. Possible values are OFF (Default), ON, Last. In 'Last', the last state is stored in non-volatile memory and this value is loaded when restarted. **This mode does not work for outputs in 'Impulse output' mode.** For more information, see the explanations after the table!
Mode	Set the working mode of a the channel For more detailed operation see the explanations after the table!
Delay[s] Impulse[s] Filter[ms]	Sets the time of the impulse in seconds for the outputs with Mode='Impulse Output'. Maximum value 16383s. *The 'Impulse' time has the precision [-100ms;0], i.e. real time is less by up to 100ms or equal to the set (from v5.62).* For digital inputs- sets the value of the digital filter which filtrates the generation of events in case of change of its condition. In this case the value is set in ms as each step is 20ms. Maximum permitted value 65535. *It is recommended to use a values of few hundred ms when it comes to mechanical contacts connected to the input- that way is ensured that there will not be many events caused by the mechanical vibration of the contacts before they manage a stable condition.*

	(from v5.59) For outputs in Mode=Manual/Toggle, a delay can be set in seconds for the execution of the switch-on commands. A value of "0" (the default) disables the delay and commands are executed directly. *The delay is useful in automatic processes and remote control where it is possible for various reasons to accumulate rapid ON-OFF-ON-OFF sequences. For some types of loads, this is not useful. The delay effectively eliminates the intermediate turn-offs and leaves only the final turn-on (after the delay time has expired). Turn off commands ALWAYS execute instantly!*
Invert Output	Checking this option leads to 'inversion' of the commands that are sent to the relay. Use it when the load is connected to normally closed (NC) contact of the relay – that way the condition ON in the interface and the SNMP will correspond to 'LOAD ON'.

The 'Default' field value has a different meaning depending on whether the channel is input or output. If it concerns an output (e.g. relay) this determines the state of the relay after the power supply is turned on. In this case the value of the parameter 'Invert Output' is important too.

For your convenience you can use the rule**: when the load is connected through normally closed (NC) contact turn on 'Invert Output'. This is compulsory if you want the cloud platform to properly read the on/off conditions of the load (respectively to calculate the consumed energy).**

Condition ("Default")	"Invert Output"	LOAD through N.O. contact	LOAD through N.C. contact	The coil of the energized relay
OFF	NO	TURN OFF	TURN ON	NO
ON	NO	TURN ON	TURN OFF	YES
OFF	YES	TURN ON	TURN OFF	YES
ON	YES	TURN OFF	TURN ON	NO

When speaking of digital input the 'Default' parameter determines how much potential to be internally passed towards the input so that when it's being left unconnected (open) to have a certain stable condition.

That potential is obtained as a result of internal 'connection' between pull-up and pull-down resistor (~50kOm) to +3.3V or 0V. This functionality allows to such input to directly connect a mechanical contact (from button,reed ampula etc.) to 0V (when Default=HIGH/ON) or to +3.3V (when Default=LOW/OFF).

In this case the 'Invert Output' setting does not affect towards which potential the input should be internally connected. But it will invert the value that is read from the input.

The 'MODE' parameter sets the working mode of the channel. There are three modes available for the outputs:

- **„Manual Output"** - manual control of the channel. The change of its ON/OFF status happens with a manually submitted command through Web/SNMP/SmartSpaceCloud.
- **"Impulse Output"** - Impulse output which after a command changes its condition for a period of time (set in the Impulse(s) column) and then it returns to the default (i.e. set in the 'Default' field) condition.

When the command is submitted through SNMP or Macros you should bear in mind that the starting of the impulse (whether the 'Default' of the output is Low or High) happens with setting the condition ON=1.

©2008-2024 www.ipnetcontrol.net

During the impulse is permissible to issue a command OFF=0 – it will forcibly return the output to its initial state.

If you submit a repeat command to start the impulse while it is already running, it will lead to the beginning of a new measurement of the time of the impulse, i.e. the releasing of the impulse starts from the beginning (it prolongs).

- „**Toggle Output**" - submission of any kind of command (ON or OFF) leads to an inversion of the output's current state. This mode can be used for a higher-speed inversion of the output that generates signals periodically (it is not suitable for relay outputs).

3.3.2 Analog I/O Channels

Different types of measuring channels (temperature, voltage, electrical current, etc.) are sent to the built-in analog-to-digital converter (ADC). It converts the voltage generated into a digital code by the sensor and after that it calculates the initial value the respective formulas.

As with the other channels here you can name every channel (**Name**) and determine whether it will be visible on the 'Status' page (**Visible**).

The drop-down menu '**Mode**' defines the dependency according to which the voltage measured by the input is converted in to the respective physics quantity.

Since the same channels could measure different quantities (according to the sensor that is placed on them) then any quantity can be set in them but this does not mean that the device can actually measure it.

For the models that have a channel for measuring main voltage from here it should be set whether it will be measured VDC or VAC.

The "MA Filter Points" field determines the number of measurements of the built-in Moving Average filter (from v5.39) for analog inputs: from 1 (no filtering) to 256 measurements. The filter averages the number measurements and thus effectively removes noise from them (at the expense of fast action). **Unlike the old filtering method, here it applies to all modules that have access to analog inputs - Automation, Web, MQTT, SNMP, SPC !!! Only in SNMP is there access directly to the analog input before the filter.**

The reaction time of the filter, ie. the time between the change of the input value and the start of Automation, for example, is determined by the formula:

Tf = N * ScanInterval,

where **N** is the number of filter measurements (1, 2 ... 256)

ScanInterval - scanning period for analog inputs (see 3.3.3)

*In older firmware versions, instead of the MA filter, there is a primitive filter similar to that of digital channels: **"Filter [ms]"**. Here it is only related to the operation of the Automation block: it determines how many ms the measured value must be stable below/above the set threshold before Automation is executed. This filtering is extremely important for the stable operation of Automation modes, because without it (or at very small values of the filter) can occur "oscillation" around the set threshold, which leads to the generation of many output commands.*

3.3.3 Other global input settings

From v5.46 the section "Miscellaneous parameters" has been introduced, in which general settings for all inputs are set.

Currently available settings are:

Analog inputs scan interval	Determines the scan frequency (ScanInterval) of the analog inputs. The step is in 20ms, the factory is 20ms, but can be increased to 2000ms.
"Toggle" only	For digital outputs that are in "Mode = Toggle Output", this check box

with 1 (0=Off)	determines whether the status reversal should be done only by passing 1/On (when checked) or with both values 0 and 1 (Off and On). In the first case, passing 0/Off acts as a standard command and the output takes this state.

In firmware v5.19 was added a new ability for the user to define up to 2 specific sensors that are connected to the analog inputs. This allows to use any type of sensors with voltage output in the range 0 ... 3.3VDC.

The two user sensors appear in the 'Mode' drop-down menu as 'User defined 1, 2'. To edit them you need to go to the link "Edit user defined sensors". For the setting it is necessary to know the conversion function of the sensor, which must be able to be described as:

U[V] = a + b * 'Sensor physical value'

, where U is the voltage at the output of the sensor (in volts);

a and b are coefficients (real floating point numbers)

'Sensor physical value' - is the value of the physical quantity that measures the sensor (eg. degrees, Bar, wind speed, etc.)

Take for example a pressure sensor 0..16Bar with an output according to the standard 0..10V. We can easily find that its function of voltage conversion is:

U[V] = 10/16 * P[Bar] => a = 0.0, b = 0.625

Enter the dimension of the measured value from your sensor in the 'Dimension' field (up to 4 characters allowed) - it will be used when displaying the values in the main page, in the Automation block and in the MQTT JSON messages. You can leave the dimension blank.

Sensor data is rounded to two decimal places. If you need more accuracy, you can rework the formula and display mili-, micro-, kilo-, mega-, etc. of physical size.

3.3.4 Virtual input-output channels (Virtual IO)

Eight virtual I/O channels have been introduced in v5.59 – these are software registers that can be assigned values [0..255]. They do not correspond to real hardware inputs or outputs. Their value is not preserved when the device is restarted and they are set to zero.

Their main purpose is to extend the capabilities of the Automation, Macros, Timers functions. And in some specialized models, they will be used to access information received from external devices (eg via RS-485).

They are accessed via MQTT (write/read), via ModbusTCP (read only as 32bit pairs) and Web: "IO Settings" menu, via the "Virtual Ports" link, you can access the current values of the Virtual IO ports, as well as change their value.

Every change in the value of a Virtual IO port causes a check in the Automation blocks in which it participates as a data source ('Main Sensor').

3.4. Macros – user-defined sequences of actions

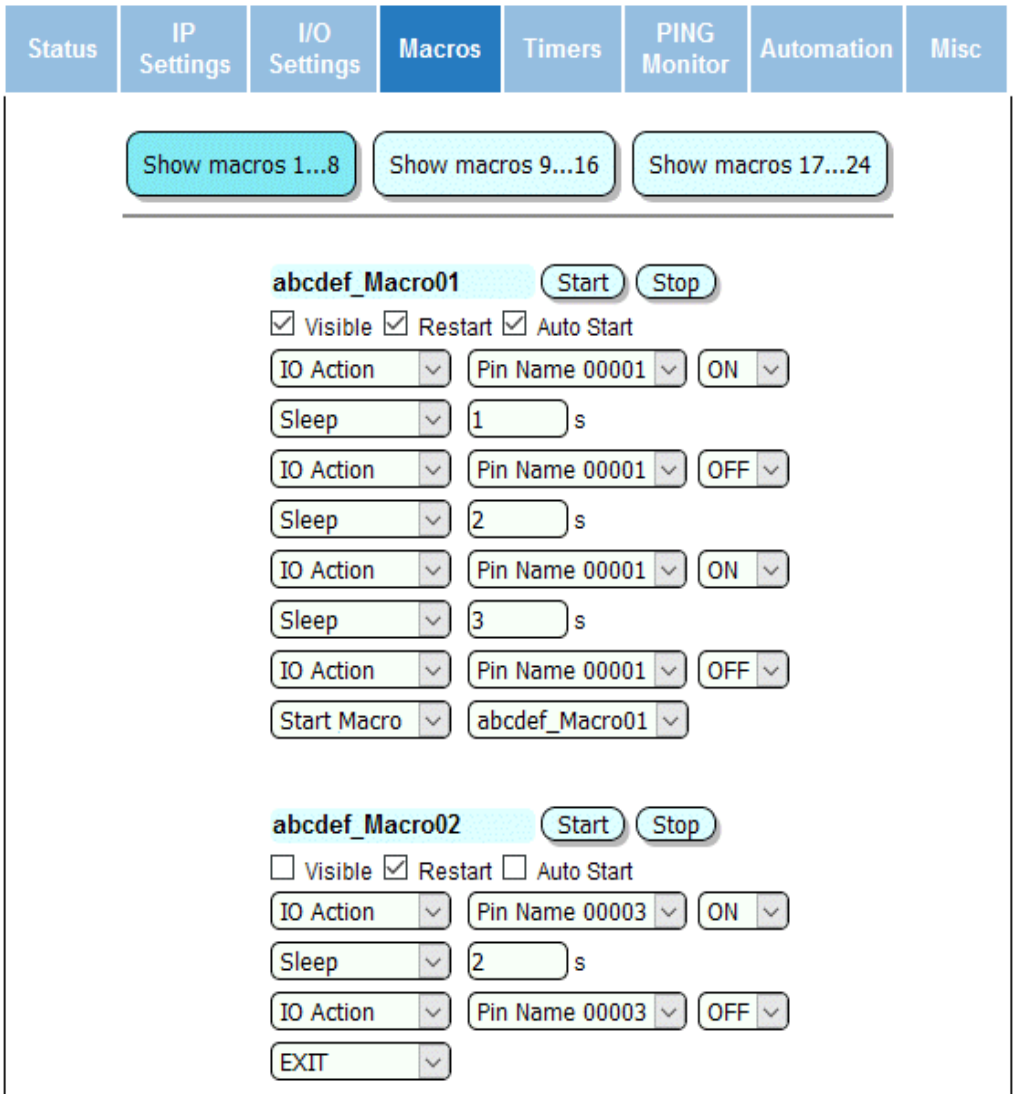
This is a new function module that allows the user to define groups of actions with output circuits combined with time delays. For example:

"Turn on Line1 → Wait 10s -> Turn on Line 2 → Wait 5s -> Turn off Line1 → Turn off Line 2 → END"

These Macros can be started manually (via Web, SNMP, MQTT), from another macro, from 'Automation', from 'Timers' or from the 'Ping Monitor' module. **These macros**

are NOT supported in the cloud platform (it has its own Automation module for this purpose).

The Macros module (accessible from the 'Macros' menu in the Web interface) contains 24 blocks for defining action sequences, each block containing 8 rows / cells.



Macros are divided into groups of 8 each for more transparency. **Please note that the Apply changes button only records changes in the current macro group that is displayed in the browser!**

The following table describes the available settings for setting each macro:

Macro name	Free text for macros name. 15 characters max.
Start / Stop	Manual start or stop buttons for a macro. Stopping is useful when working with cyclic macros.
Visible	Determines whether a button to manually start/stop the macro is displayed on the 'Status' page

Restart	Macros, with this option checked, begins execution from the beginning, every time a command / event arrives for it, no matter if the macro is currently running or not. Macros without this checkmark - do not restart from the beginning if they are currently running. **This parameter does not affect the case when the macro restarts itself and such self-start is always allowed.**
Auto Start	In this mode, the macro starts automatically when ***NetControl*** starts.
Rows/cells	Each cell (8pcs for each macro) can be: – 'EXIT': end of current macro (the current macro also terminates when the last cell is reached, regardless of what action it performs) – 'IO Action': action to an output channel (eg Line1 = ON). **Impulse outputs are always started with ON!** – 'Sleep': time delay in seconds (1 ... 65535) – 'Start Macro': Launch a macro and continue the current one. **Current one is first continued and then the new start will appear.** – 'Stop Macro': stop execution of selected macro – "Remote IO Action": send command to another device via SNMP protocol (from v5.22) – 'EXIT IF' - exit the macro if an input/output/virtual channel has a certain value at the time of execution. – 'Skip next step IF' - skip the next step of the macro if a given input/output/virtual channel has a certain value at the time of execution. – 'IF Value' – not available for user selection. It is automatically activated in combination with the previous two types of steps and sets a value to compare with. *The 'Sleep' time is set in seconds and the deviation of the value is (-100ms; 0], i.e. real time is less than up to 100ms or equal to the set time (from v5.62).*

The ability of one macro to start another allows an arbitrary amount (or even all) of the macros to be chained together as one macro with more cells.

It is also possible for a macro to restart itself: this results in an endless looping of the macro or group of macros (until their configuration is changed or stopped manually). The combination of a loop macro with **'Auto Start'** mode enables ***NetControl*** to permanently executes a sequence of actions.

"Stop macros" allows interrupting a macros by command or even as a result of Automation, Timer or Ping Monitor (since they can only start a macros, it is necessary to create intermediate macros to implement 'Stop'). Stopping a macro is a very useful control of cyclic macros: for example, switching from one cyclic mode to another.

The use of a cyclic macro group should be planned very carefully, since many different situations are created, especially when combined with Timers, Automation or Ping Monitor.

Manually stopping a cyclic group of macros will be quite difficult in practice because of the lack of information about which macros are currently active in order to pass a 'Stop' command to it. The option is to submit Stop sequentially to all participating macros, or temporarily remove the self-start and wait for the sequence of actions to complete.

Macros cannot be configured through SNMP, but can be manually started or stopped with commands:

Start: >snmpset -v1 -c private 192.168.1.100 .1.3.6.1.4.1.19865.2.3.6.M.0 i 1

Stop: >snmpset -v1 -c private 192.168.1.100 .1.3.6.1.4.1.19865.2.3.6.M.0 I 0
where M = [1..24] is the macro number

Macros can also be started or stopped via MQTT – see chapter 5.4 for more information.

You can combine Macros with “Impulse Output” outputs - this will give you even more options for combining and getting different action sequences.

3.4.1„Remote IO Action” - submition of action to another NetControl

"Remote IO Action" gives you a very interesting opportunity to transfer an automation over a network connection and control another ***NetControl*** device. Of course, this is easily achieved with the cloud platform, but this feature allows you a "peer-to-peer" communication between two devices without the presence of intermediate systems.

Depending on the model of the other device, you need to find which [P] channel numbers are available in it and select them in the macro (all possible 24 output channels are displayed, but each model has a different part of them).

The settingc for the remote device is in the *IP Settings-> SNMPv1 traps / remote IO settings tab*. These settings must correspond to the SNMP settings of the other device (the password must match).

When using "Remote IO Action" in Macros that has the "Auto Start" option (start with device power on) always leave before it a Sleep of 5-10 seconds. Otherwise, the "Remote IO Action" command is executed at a very early stage, where network protocols (eg ARP) have not yet initialized, and the command will not be sent.

Note that commands are issued over UDP via the SNMPv1 protocol. At the current level of network communications, packet losses are extremely low, even negligible. But theoretically it is possible data loss and consequent non-execution of the given command. Commands are NOT verified in any way whether or not they were received on the other device.

3.5. Timers – starting of a Macro at specified hour:minute (with DOW and month options)

The 'Timers' menu allows you to define 16 different times (hour: minute) to start a macro. Additionally a day of week and month can be specified in every timer.

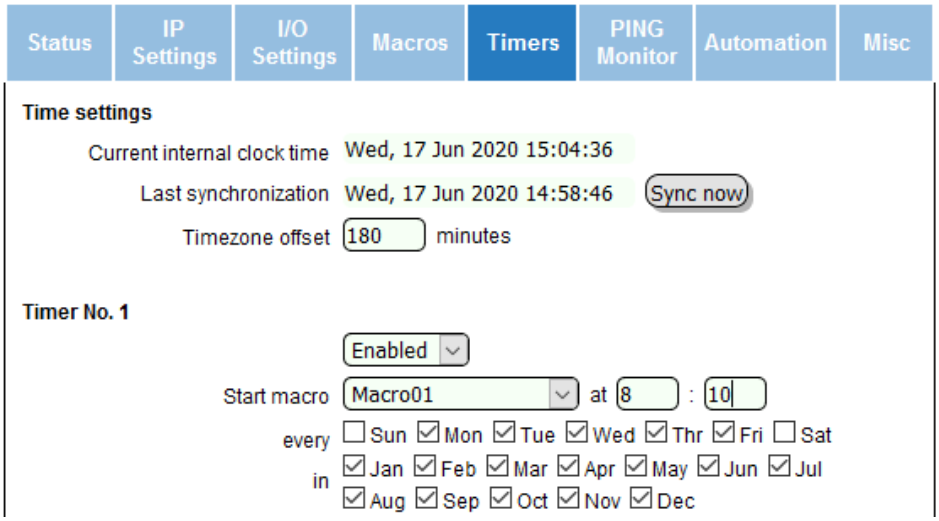
3.5.1 Synchronize the NetControl clock via SNTP

NetControl does not have a battery-backed real-time clock, so timing requires synchronization of its software clock via the network SNTP protocol (this is a subclass of the NTP protocol at UDP port 123). This protocol is supported by all public NTP TimeServers, which you can view at https://www.pool.ntp.org. Other well-known NTP servers are ***time.google.com***, ***time.windows.com***. PING to the server of your choice and enter the obtained IP address in the menu "IP Settings" -> SNTP server IP address.

The best option is to use a local NTP/SNTP server to synchronize your ***NetControl*** devices, as this way you are not dependent on external services and Internet connectivity. In addition, an intermediate server allows you to reserve the source of time, as they can poll a lot of public servers. The NTP server is built into Windows 10 (but is not enabled by default) and is easily accessible for Linux operating systems (ntpd).

Each time you restart ***NetControl***, every 2 hours and when you run the "Sync Now" or "Apply Settings" command from the "Timers" menu, a request is made for the current time to the specified server. If you do not receive an immediate response from it, the request begins to repeat periodically (initially faster, and in the permanent absence of a response - every 5 minutes).

©2008-2024 www.ipnetcontrol.net



The ***NetControl*** clock does not support winter / summer time. The user is required to specify his current offset from UTC and enter it in the "Timezone offset" field (the value is in minutes and can be a positive or negative value). When changing daylight saving time - this value must be manually adjusted.

If the network and time parameters are set correctly, your current time will be displayed in the "Current internal clock time" field.

When using DHCP, you can use the "option ntp-servers" parameter to dynamically set the NTP/SNTP server address. Note that DHCP allows multiple IP addresses to be set, but ***NetControl*** only takes the first one.

The accuracy of the software clock is not great and is not guaranteed. Therefore, it is a good idea to make sure that you have set the address of an accessible and stable NTP server. ***NetControl's*** *clock will run indefinitely after a single adjustment of its clock, but it will quickly accumulate a measurement error (tens of seconds of day shift can be expected).*

When synchronizing the time after a long pause (lack of connection, non-functioning server, etc.) there is an algorithm for tracking missed/duplicated time events (due to the hopping clock forward / backward). Therefore, it is possible to start several Times together in the new synchronization. If the difference between the current and the new time is more than 1 hour - the internal module for monitoring the timers is re-initialized and starts from the beginning. In this case, events that have been in the "hole" of 1 hour will be lost or duplicated (depending on whether the internal clock is lagging behind or speeding up).

The Timers module actually starts working after the first time synchronization. If this does not happen at all - the module will not create any event. However, if it is synchronized once - it will work indefinitely, regardless of whether or not there are new synchronizations via SNTP (respectively, an unlimited clock error will accumulate). Setting the IP address 0.0.0.0 for the SNTP server is equivalent to disabling the entire module with Timers (the settings of the individual timers will be saved). When deactivated or not connected to the server, "Not syncronized" is displayed in the "Current internal clock time" field.

©2008-2024 www.ipnetcontrol.net

3.6. 8 (24) channel ‘PING Monitor’

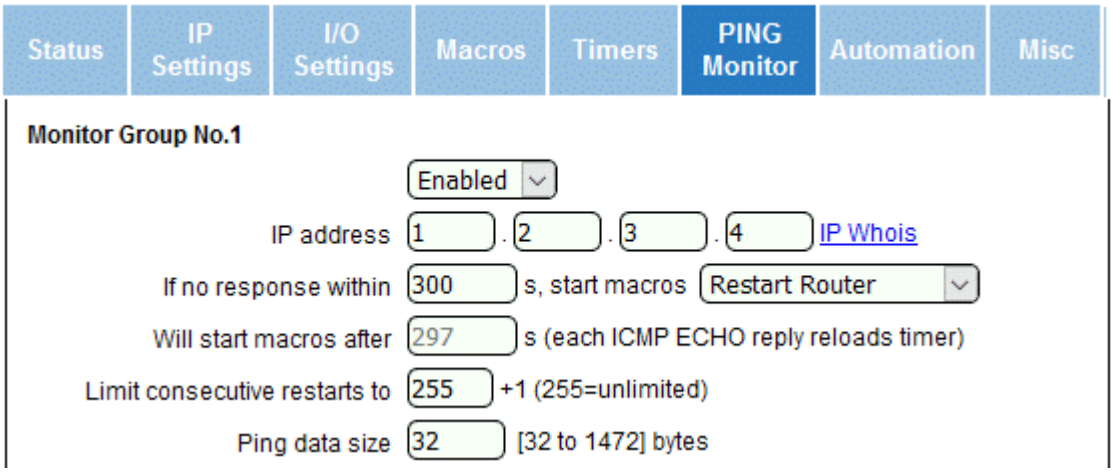
Since one of the main applications of the ***NetControl*** series is active monitoring of the network paths and restart of the locked equipment, we developed a multi channel software module for PING towards up to 8 IP addresses. For 24R3S2A model the ping channel number is increased to 24. If PING is lost, a macro can be started to restart the equipment through which the broken link goes.

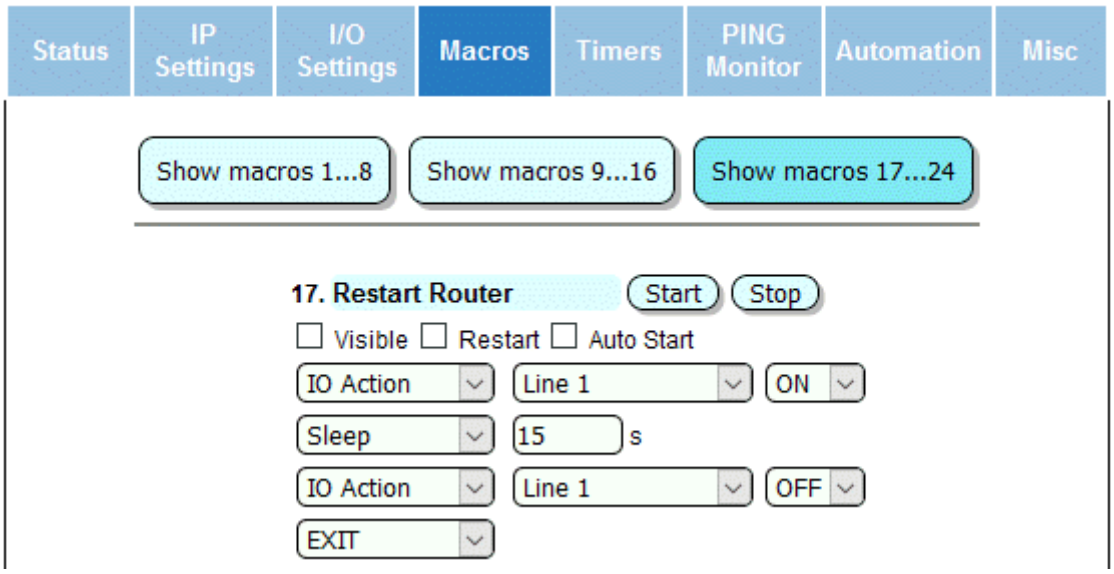
In the ‘PING Monitor’ menu you have an access to all the ‘**Monitoring Group No.**’. They have the same settings parameters:

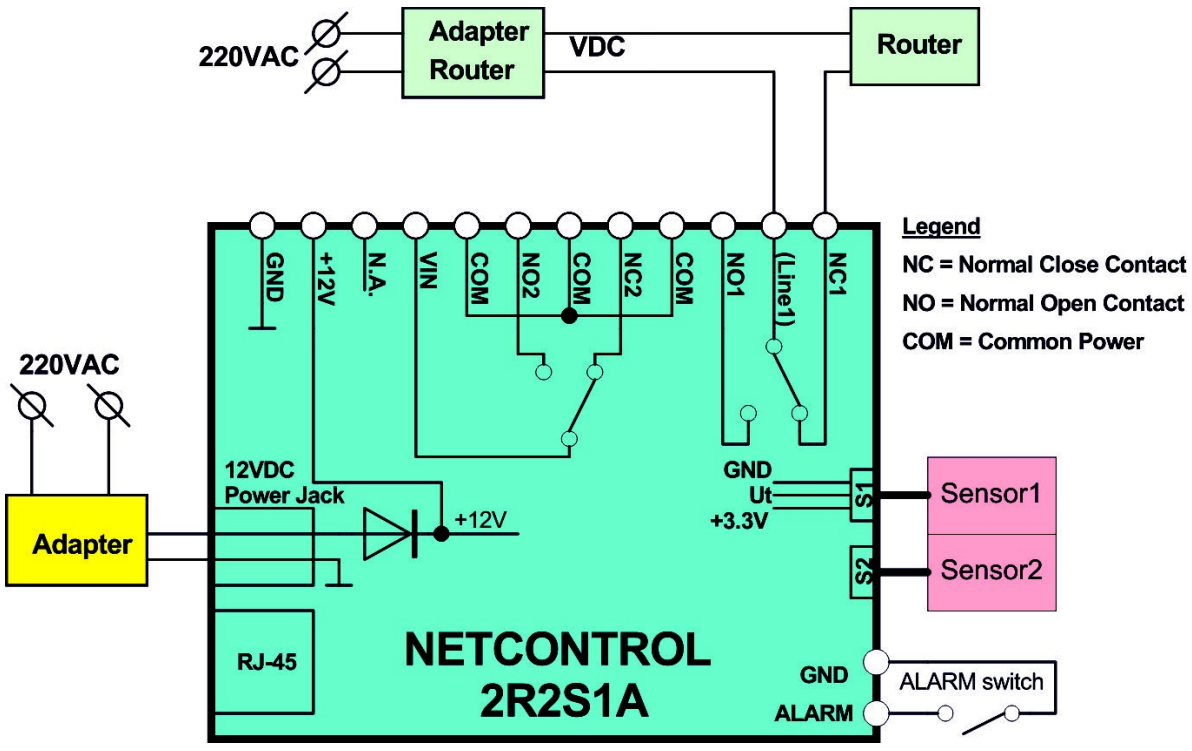
Enabled/ Disabled	Determines whether the group is enabled or not. Actually the disabling of a group is equivalent of setting a zero value of „**If no response within**„
IP address	Address to which the ICMP Echo requests will be sent.
If no response within	Determines the amount of time during which if there is no response from the IP address, it is considered that there is no connection. The permissible values are from 40s to 16383s.
start macros	Which macro number to run if the IP address is lost. None can be selected - no action will be taken in case of loss of connection, but when combined with 'ACT Led mode' it can be used for visual diagnostics.
Will timeout after	This field shows in how long the period after which we consider the connection severed will expire and macros will be started. Since the requests for each group are submitting in about every 10-16s it is normal here to reach values lower than the set time in ‘**If no response within**’ and after that the counting starts from the maximum. *This field can be used for feedback whether there is PING to a certain address: you can choose to=Nothing and to monitor whether ‘**Will timeout after**’ is recovering every 10-16 seconds. If it’s not recovering and it is steadily decreasing toward 0- then the device does not receive ICMP Echo responses i.e. ‘There is no PING to the set address.’* *The page does not reload automatically; periodically you should select ‘PING Monitor’ menu in order to see the current value of this field.*
Limit consecutive restarts to	In case of a permanent connection loss from here you can limit the number of the consecutive commands that are sent towards the channel for response (if there is no need to endlessly restart an equipment which does not have a problem)
Ping data size	Defines ICMP Echo request data length. Allowed values from 32 to 1472 bytes.

There may be situations where the PING monitoring period is shorter than the run time of the macro (or group of macros). Then a new macro startup will be generated and the behavior will depend on the settings of the macro itself (if the 'Restart' option is enabled or not).

The following two images show sample Ping Monitor and Macros settings that can restart Line1 for 15s when Ping is loss for over 300s to address 1.2.3.4. Also is shown how to connect a normally closed relay contact from ***NetControl*** to the router's power supply so you can restart it.







©2008-2024 www.ipnetcontrol.net

3.7. Automatic tasks (menu ‘Automation’)

You have access to 8 different automatic tasks which generally allows you to create an action towards some of the outputs via Macros (and to send SNMP Trap/MQTT/SPC message) in case of a change of some of the analog inputs outside of the preset limits. There is no limit at how many times a certain input or output will take part in automatic tasks but it depends on the user there not to be mutually incompatible actions.

Value compare mode	Determine the comparison mode of the input value. For more information see the explanations below the table.
Thresholds	Lower (LOW) and higher (HIGH) threshold of the comparison functions of the input quantity. The set values correspond directly to the dimension of the selected 'Main sensor' (and to the range of the quantity it measures).  *The values entered by the user are recalculated according to the resolution of the analog-to-digital converter and can therefore be adjusted automatically to the nearest discrete value.*
Main sensor	Analog input channel whose value will be compared. Channels are displayed with their names and their dimension. (from v5.59) In addition to the analog channels, you can also choose from the Virtual IO channels as a source of information. These channels are "checked" for Automation events EACH TIME THEIR VALUE CHANGES.
Diff. sensor	Second analogue channel to form a differential mode. The value of (Main - Diff. Sensor) is actually compared. Can not be used with Virtual IO channels. **The differential mode is deactivated by selecting 'Not used' in this field.**
On Event	Which macro to run when satisfying the set "Value compare mode" mode. SNMP trap message is ALWAYS generated, and this can be selected to remain the only response. Only macros 1 through 8 (inclusive) are allowed.
On Restore	This points the macro that will start when Hysteresis (ACC) mode value is restored. Recovery macros are from 9 to 16 (incl.) and cannot be selected by the user.

In regard of the monitoring of the analog input quantity the following modes of operation (‘**Value compare mode**’) are available:

Disabled	The whole automation task is disabled
<LOW	The 'On event' macro is STARTED ONCE when the measured value falls below the set LOW threshold. Nothing happens on recovery except that you are given a new permission to start 'On Event'.
>HIGH	The 'On event' macro is STARTED ONCE when the measured value falls above the set HIGH threshold. Nothing happens on recovery except that you are given a new permission to start 'On Event'.
<LOW or >HIGH	The 'On event' macro is STARTED ONCE when the set HIGH

©2008-2024 www.ipnetcontrol.net

	threshold is exceeded or when the measured value falls below the LOW threshold. Nothing happens at a value in the interval [LOW; HIGH], except that you are given new permission to start 'On Event'.
Hysteresys (ACC)	The 'On event' macro is STARTED ONCE once the HIGHthreshold is exceeded. When the measured value falls below the set LOW threshold, the 'On restore' macro starts.Nothing happens at a value in the interval [LOW; HIGH]

Regardless of the chosen mode of work is necessary that the input quantity keep the value over/under the thresholds for the minimum amount of time of the filter that is set for the analog input in the menu '**I/O Settings**'. If the measured value fluctuates around the threshold no action will be generated until it permanently (or at least for the time the filter needs) level the inequality.

The modes **'<LOW', '>HIGH', '<LOW or >HIGH'** function in a similar way. The implementation of the inequality and of the filter of the input leads to starting of the 'On Event' macros. This action is performed ONLY ONCE and can be re-issued only if there was a moment when the inequality was not satisfied. The modes are suitable to use in situations requiring action type 'Alarm'.

The generation of the SNMP trap message in these modes operates as follows: message with input value is sent after the first leveling of the inequality and the filter. If the inequality continues to be satisfied permanently – no more traps will be sent. If in the next moment the value of the analog input ceases to level the inequality (even once) – then sending of message is again permitted (when the conditions are met again).

Everything said so far proves that the choice of the value for the filter is very important for obtaining optimal results! The rule is: The higher the value is –the more stable will be the work of the automatic algorithms but at expense of their promptness.

"Hysteresys (ACC)" mode works as starting 'On Event' macro when the upper threshold is exceeded and starting the 'On Restore' macro when value below the lower threshold. It is suitable for outputs in Manual Output mode. This is practically the most common way of working to realize:

- **Thermoregulation with heater**: relay output turns on when the temperature drops under the set threshold and it turns on after reaching the set temperature
- **Thermoregulation with ventilation**: analogical to the heater but with a reverse status of the output circuit (the relay)
- **Managing the charge of the accumulator**: turns on the charged circuit when the voltage of the accumulator is under the set threshold and turns off the charge when it reaches 14.4 V

Here, SNMP Trap message is sent once when passing the value above the HIGH threshold for the filter time. Message is then generated when the value falls below the LOW threshold. Nothing happens in the neutral zone (LOW; HIGH).

Each SNMP trap contains the value of the analog input (as a 12bit integer) at which the inequality was satisfied.

Keep in mind that SNMP traps are processed over a period of 2s, ie. if the analog filters are small, events can be generated faster than processed and sent as an SNMP trap. In such cases, new

events overwrite the old ones and as SNMP trap is send what was last left at the time the information was processed.

For analog inputs in "ContactSwitch (Alarm)" mode, in addition to the SNMP trap, a message is generated in the MQTT / SPC channel to inform services for alarm change.

Automation blocks with Virtual IO as source DO NOT generate SNMP trap messages!

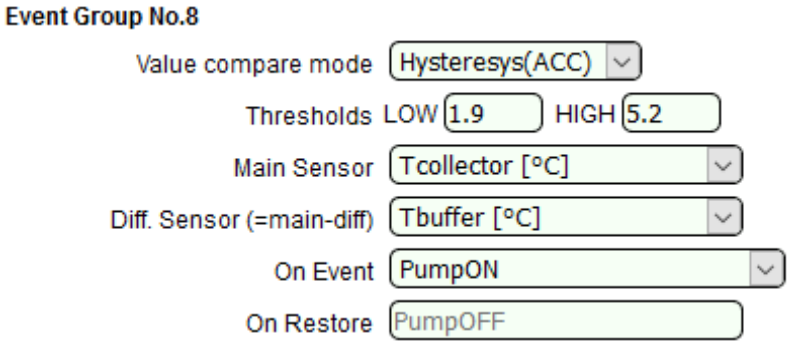
An interesting combination occurs if you use this mode with an output in 'Impulse Output' mode or macros doing the same thing – then the output will restore (after the time of the impulse output expires) even if the input parameter does not drop under 'LOW' but is in the neutral zone. That way we will get 'protective time' and that would not permit the output to stay in the set status for an indefinite amount of time.

3.7.1 Differential measurement mode

Very often, in practice, it is not necessary to monitor the value of one sensor (eg for temperature) but the difference between the values of two sensors. One example is the task of controlling a circulation pump in solar hot water systems: the pump is activated only when the difference between the temperature of the collectors and that of the buffer / boiler exceeds a set value; the pump stops when the water temperature in the boiler approaches that of the collectors.

Let's see how this mode can be adjusted to the following image:

1) Obviously, the solution in this case is the HYST operating mode, which will give the pump the most stable operation.
2) We select the two temperature sensors: 'Main Sensor' and 'Diff. sensor ' **The device will now use the 'Main-Diff' difference as a value for calculations.**
3) We set the upper (5.2°) and lower (1.9°) threshold of difference **(positive values, such as LOW<HIGH)**, the algorithm of operation follows the following inequalities:
   If (Main-Diff) > HIGH → 'On Event' macros
   If (Main-Diff) <= LOW → 'On Restore' macros
4) We select pre-set macros that turn the pump on and off.



The following table shows the process step by step: in the initial state the boiler is 35°, the panels 37° - there is not enough temperature difference to start the pump. Once the panel reaches 41°, we have above 5.2° difference - the pump switches on. From that moment the temperature of the water in the boiler starts to increase. The pump runs until the difference is very small at a boiler temperature of 40° (the difference is already <= 1.9°).

It can be set to LOW=0 ° and then the pump will run until Tbuffer equals or exceeds Tcollector (in this particular case, it is not possible to exceed the solar temperature, but in other situations it is possible). Since the equality of two sensors is practically difficult to achieve (due to noise, measurement inaccuracies, etc.) it is better not to rely on pure

equality but to use LOW>0 (practically circulation at negligible temperature difference may be meaningless because of the small heat gain at the expense of electricity).

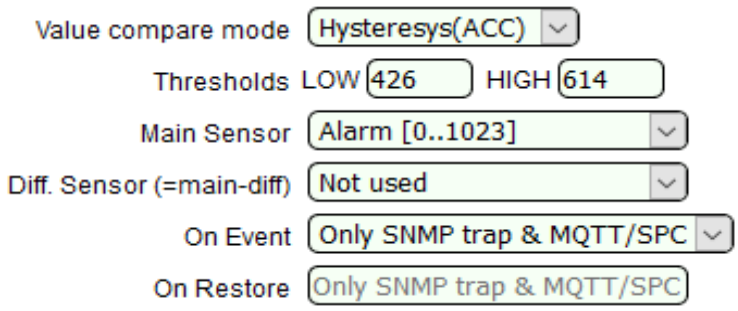
Main, °C	Diff, °C	Main-Diff	Действие
37	35	2	няма
40	35	5	няма
41	35	6 (> 5.2)	'On Event' (вкл. помпа)
41	36	5	няма
41	38	3	няма
41	40	1 (<1.9)	'On Restore' (изкл. помпа)

In the menus there is no limit to selecting only one type of differential sensors (temperature, humidity, etc.). Threshold dimensions follow that of 'Main sensor', with differential measurement also going to work with mixed sensors, but it is necessary to manually recalculate the thresholds relative to other types of sensors.

Differential mode can also be used in LOW, HIGH and LOW / HIGH modes.

Factory setting to generate alarm input events to external managing service

You will notice that every ***NetControl*** by default has one Event Group predefined. This is Event Group No. 8 (see the picture below). This automation task is needed to allow alarm input of the device to signal immediately every change in its state to the cloud platform or MQTT. As the alarm input is implemented via analog input channel it doesn't have other ability to send event on change in state (it will only send its state to the platform periodically just like other analog inputs – temperature, voltage, etc.)



3.7.2 Running macros from digital inputs (from v5.46)

In some models, in addition to the standard analog inputs, digital ones are also available ("Alarm" type). In order for a change of digital input to lead to macro start, the section "Macro start from event on digital inputs" is provided. This section is only displayed if the device has digital inputs!

Note that the digital input-macro number connection is fixed (macros cannot be randomly selected) and can only be activated or not. Also keep in mind that macros 9 to 16 can participate in Automation blocks and should be planned so that there is no duplication with digital inputs.

If it is necessary to change the level of the input signal that will start the macro, you must use the "Invert" option in the input settings.

©2008-2024 www.ipnetcontrol.net

3.8. Restarting, software updating, etc (menu 'Misc')

3.8.1 User/password for Web users

From here you can set your own username and password. The only requirement is that their length is **4 to 12 symbols in LATIN CHARACTERS**!

From firmware v5.35 it is possible to define an administrator user and an "IO User" with access only to the home page with the status of the input-output circuits and macros. If "IO User" is in "Action" mode (see menu "IP Settings-> IO User web account") it can be used to send commands to the outputs of the device or to run macros. Otherwise, he can only see the current states.

There is a "Logout" link at the bottom on the main page for switching between users.

3.8.2 Restore factory settings

You have the option to do two separate resets of the settings: one of the default settings of the IP part, one for the input/output circuit settings and one for channel names.

3.8.3 System software update via TFTP

The device is equipped with TFTP client that after a command sent from the button '**Start Firmware Update via TFTP**' (or via SNMP) connects to the IP address of the TFTP server and downloads (if available) the necessary file with the update. After the completion of the update the device restarts on its own.

It is highly recommended not to do the updating of the system software in real conditions (large networks, remote power supply, etc.). Failure of the power supply during the updating of the system software will lead to damage in the device!

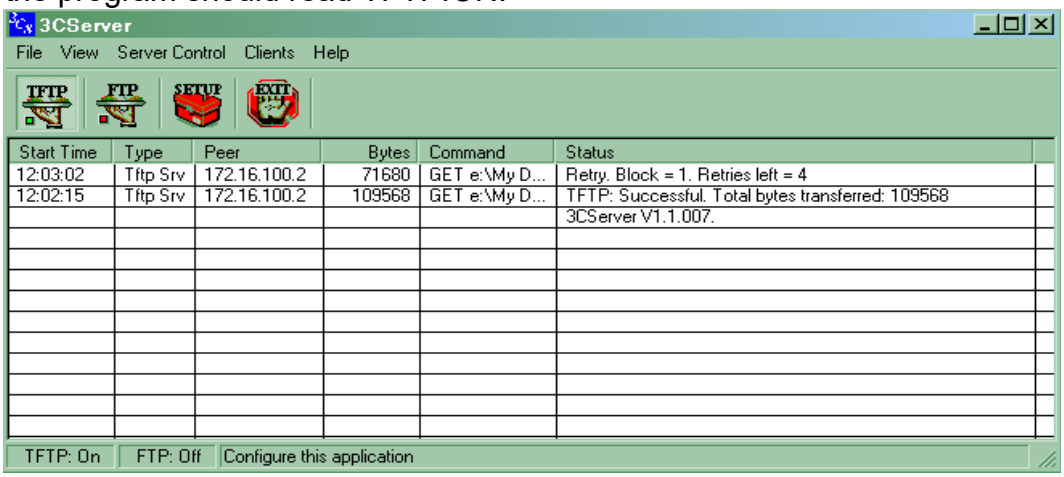
©2008-2024 www.ipnetcontrol.net

*The return of **the older version** (so called **downgrade**) in most of the cases will require once a follow up loading of **the factory settings**. This also means that the process should not be done in real conditions (remotely).*

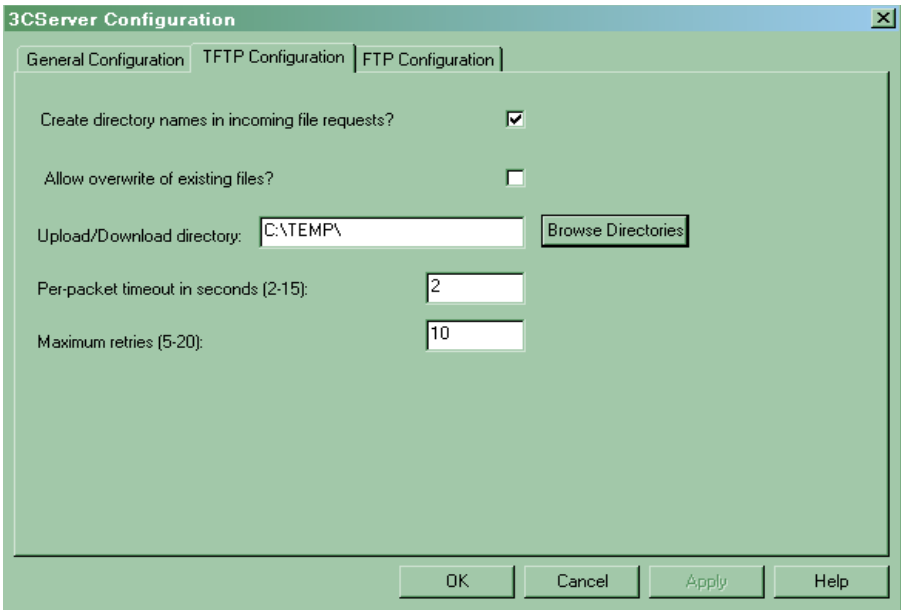
It is most suitable to update the software of a device that is loaded with factory settings.

For the correct progress of the updating process you must go through the following steps:

1. Install the program '3CServer' (download here) за Windows. Other programs can be used too, including the ones embedded in Linux TFTP daemons.
2. Start the program, use the 'TFTP' button to start the server. The status bar of the program should read TFTP:ON.



3. Go to the SETUP of the program then to TFTP Configuration



It is important to see/set 'Upload/Download directory' and to put the file for the updating 'pipfw5.bin' there. **The file must be in the directory itself and not in subdirectories; its name and extension should NOT be changed.** Set the parameters for timeout and retries as it is shown.

4. Make sure that installed Firewall programs do not block the application.
5. The device must have authorization to update via TFTP ('TFTP firmware update=ENABLE'), correctly set IP address of the respective TFTP server and correct network settings.
6. Start the update- via WEB or SNMP.

©2008-2024 www.ipnetcontrol.net

DO NOT TURN THE DEVICE OFF DURING THE UPDATE PROCESS!!!

7. In the status window you should see the information for the file transfer with the amount of the already transferred information. It is normal to receive messages for 'Timeout' and 'Retry' in some portions of the data- this should not worry you if after that the transfer of the data continues again on its own.
8. Once the transfer is complete the device will restart. IMPORTANT!!! The last package of the TFTP file will not be confirmed by the device and that's why the server will try to send it several times and after that it will mark it as an error. This should not worry you, the device has accepted the last part but as it is restarting now it will return the response to the server after that.
9. You will be redirected to the 'IP Settings' page where you should be able to see the new version already. When the access is done via Web make sure that you load 'fresh' versions of the Web content- better yet clear your browser's cache.

When there is a lack of connection to the server the device makes several attempts to connect before it finally ends the execution of the command without updating. During the update you may notice disturbance of the operation of the various processes in the device.

3.8.4 System software update via domo.ipnetcontrol.net

It is provided the possibility for automatic updating of all devices that are registered in the cloud platform. That way is not necessary for the user to watch all the time for the newer versions (to install the TFTP server, etc.) – they will be automatically uploaded on the connected devices (that requires your device to be ONLINE on the platform).

When a newer version of the software is posted you will be notified on the platform and you would be able to confirm the update.

3.8.5 Update system software via Web (from v5.38)

The ***NetControl's*** system software can also be updated by uploading the * .bin file via the 'Misc' page of the embedded Web interface.

After starting the file upload, wait for a message to be displayed for the process to complete successfully (no real-time status).

In this way the software of the WiFi module cannot be updated - it is possible only via TFTP.

3.8.6 Saving and restoring the configuration to/from a file (from v5.34)

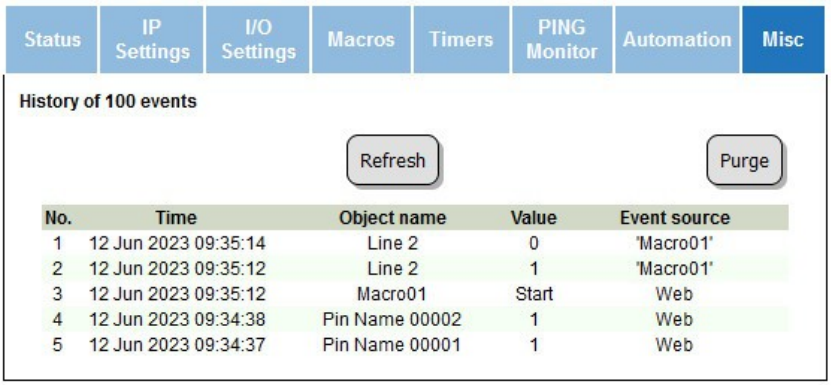
In the "Misc" menu you can also use the "Configuration" section with the "Backup" and "Restore" buttons. With them you can save in a file the entire current configuration of the ***NetControl*** device - this covers absolutely all settings - IP, IO, Names, Macros, Ping Monitor, Automation, Timers. The created file can then be loaded on other devices and thus save setting the same parameters in each of them.

3.8.7 "Events Log" - log of last 100 events (from v5.55)

From here you can trace all outgoing IO events, start of macros and timers. When the clock is set (by SNTP), the date/time of the event is also printed (otherwise 'n/a' will be printed). The "Event Source" field displays information about the source of the event; for example for relay output - Web, SNMP, MQTT, SPC.

The buffer is for 100 events. After filling it, new events are not registered, but with the "Purge" button you can clear them and everything newly received will automatically start to be recorded.

©2008-2024 www.ipnetcontrol.net



No.	Time	Object name	Value	Event source
1	12 Jun 2023 09:35:14	Line 2	0	'Macro01'
2	12 Jun 2023 09:35:12	Line 2	1	'Macro01'
3	12 Jun 2023 09:35:12	Macro01	Start	Web
4	12 Jun 2023 09:34:38	Pin Name 00002	1	Web
5	12 Jun 2023 09:34:37	Pin Name 00001	1	Web

In v5.58, an option was added for another mode of operation of the EventLog buffer - 'Circular Buffer' (activated from the menu 'IO Settings->Miscellaneous parameters'). In this mode, the buffer always contains the last 50 events (new ones drop the oldest) and does not need to be cleared by the user.

©2008-2024 www.ipnetcontrol.net

4. SNMPv1 access

4.1. Access to I/O through SNMP

As it was already mentioned the device supports access to all parameters and functions through SNMP. In SNMP every object is characterized with a specific sequence of numbers (OID) or symbol name (only if the MIB files are correctly installed to the SNMP client).

In this section are described all objects that are directly related to the management/access to the input/output circuits. All other objects you can find in section 8, where the tree structure of the objects included in the device is depicted.

4.1.1 SNMP objects for personal access to the input/output channels

In the MIB structure of ***NetControl*** is present a group of objects 'iop1'…'iop32' which corresponds to all systematically available for the kernel input/output circuits. Not all of them are displayed in a specific model of the device but the objects are present nonetheless.

In each of these objects which OID is formed as .1.3.6.1.4.1.19865.2.3.1.P. where [P] is the number of the channel from 1 to 32 are present few sub-objects which could retrieve/record information of every channel:

MIB name	Digital OID	Description
ioNum[P]	.1.3.6.1.4.1.19865.2.3.1.[P].1.0	Number of the channel [1, 32]
ioName[P]	.1.3.6.1.4.1.19865.2.3.1.[P].2.0	Name (only read!)
ioMode[P]	.1.3.6.1.4.1.19865.2.3.1.[P].3.0	Mode
ioDefault[P]	.1.3.6.1.4.1.19865.2.3.1.[P].4.0	Default value
ioImpulseFilter[P]	.1.3.6.1.4.1.19865.2.3.1.[P].5.0	Impulse/Filter
ioValue[P]	.1.3.6.1.4.1.19865.2.3.1.[P].6.0	Value (R/W)
ioReadAll[P]	.1.3.6.1.4.1.19865.2.3.1.[P].7.0	Read all
ioInvert[P]	.1.3.6.1.4.1.19865.2.3.1.[P].8.0	Invert parameter for outputs
ioGauge[P]	.1.3.6.1.4.1.19865.2.3.1.[P].9.0	Get channel value as 32 bit unsigned integer. Use this object for pulse counters! It is recommended to use this object when reading channel value instead regardless of the channel type. The object is read-only!
ioPulseCfg[P]	.1.3.6.1.4.1.19865.2.3.1.[P].10.0	See section Error: Reference source not found
ioValueFiltered[P]	.1.3.6.1.4.1.19865.2.3.1.[P].11.0	Value after MA filter (v5.39)

The parameters correspond to ones available for configuration and already described in the previous sections parameters from the 'I/O settings' WEB menu.

Deliberately from now on in the examples we use only the digital OIDs because they always work regardless of whether you have installed the MIB file or not! Of course if the file is correctly loaded in the SNMP client it could have access directly to ioValue9.0 – the value of channel 9 (i.e. Line1).

©2008-2024 www.ipnetcontrol.net

The following table shows the connection between I/O channel from different models and its number in the SNMP structure:

Name of the channel	Number [P]	Access	Notes
Line1	9	R/W	0 (Low) = the relay is off (NC circuit is CLOSED) 1 (High) = the relay is on (NC circuit is OPEN)
Line2	10	R/W	
Line3	11	R/W	
Line4	12	R/W	
Line5	13	R/W	
Line6	14	R/W	
Line7	15	R/W	
Line8	16	R/W	

For example if we want the state of Line4 to be ON we must complete a SNMP command which should record ON=1 in the object ioValue of channel 12:

> snmpset -v1 -c private 192.168.1.100 .1.3.6.1.4.1.19865.2.3.1.12.6.0 i 1

4.1.2 Other (general) SNMP objects for access to the input/output circuits

Beside the described group of objects that have the same structure for all available channels there is one more group which is placed under the branch of '**ioMisc**'.

MIB name	Digital OID
iomReadAllOld .1.3.6.1.4.1.19865.2.3.2.1.0	**IMPORTANT: Do not use with new applications. It is retained with the purpose of reverse compatibility.** Reads the values of the all input/output circuits in the format of the ***PicoIP*** device (older firmware for ***NetControl***). No access to the all 32 objects! The format is: 'P3,P5,P6,P6.1,P6.2,P6.3,P6.4,P6.5,P6.6,P6.7,P6.8' All values are hexadecimal numbers, for example: 0x40,0x80,0x00,0x0055,0x00BD,0x00AA,0x008D,0x005C ,0x0045,0x003E,0x0049
iomReadAllHex .1.3.6.1.4.1.19865.2.3.2.2.0	**Do not use this object when you have pulse counters as they are 32bit!** Reads all channels (32), as the result is hexadecimal array (Hex-STRING) with two bytes for a channel (MSB first). For example: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 01 00 00 00 00 00 00 00 00 00 00 00 D8 00 14 00 06 00 38 00 39 00 41 03 FD 01 7C
iomSetAll .1.3.6.1.4.1.19865.2.3.2.3.0	The record of 0/1 (Off/On) in this object leads to applying this state to ALL output channels. It is equivalent of the commands All On", "All Off" in the WEB interface.

©2008-2024 www.ipnetcontrol.net

4.1.3 Reading of the temperature (sensor TDS300) via SNMP

The object for the temperature does not return the value of the temperature directly but the value of the analog-to-digital converter (this is valid for all analog inputs).

For reading of the temperature input of the ***NetControl*** series you have to use the command:

> snmpget -v1 -c public 192.168.1.100 1.3.6.1.4.1.19865.2.3.1.25.6.0

The received value, for example INTEGER: 217 must be converted in degrees (Centigrade) by the formula:

t[C°] = (3300*(SNMPValue/1023) – 500) / 10

4.1.4 Reading relative humidity (sensor HDS300) via SNMP

The humidity sensor uses the same input as the temperature one. Thus the snmp command for reading data is same is shown in section 4.1.3.

The received value is converted to RH% with the help of the formula:

RH[%] = 125*(SNMPValue/1023) – 6

4.1.5 Input for measuring the Unet (VIN)

The command for reading of the voltage Unet is:

> snmpget -v1 -c public 192.168.1.100 1.3.6.1.4.1.19865.2.3.1.26.6.0

Again the received value should be converted by a formula which depends on the type of the voltage and power supply:

- when the Unet voltage is alternating (AC) and the supply is 12V with direct connection (SMPS K0C6P with rectifier on the input):

Unet [VAC_{RMS}] = 3.3*(SNMPValue/1023)*37.25 + 0.8;

- when the Unet voltage is alternating (AC) and the supply is 12V with galvanic isolation (SMPS iK0C8P with rectifier on the input):

Unet [VAC_{RMS}] = 3.3*(SNMPValue/1023)*37.25 + 0.4;

- for DC voltage:

Unet [V_{DC}] = 3.3*(SNMPValue/1023)*49+0.4;

- for DC voltage: (0...62VDC) for 4RU1SH2S:

+Uin[V_{DC}] = 3.3*(SNMPValue/1023)*19.2927;

4.1.6 Alarm input

The status of this input is received with:

> snmpget -v1 -c public 192.168.1.100 1.3.6.1.4.1.19865.2.3.1.31.6.0

When the contact of the alarm input is ‘open’ the returned value is close to the maximum 1023 (can be with up to 10-15 units less).

When the contact of the alarm input is ‘closed’ the value drops under 10 units.

4.1.7 Input for measuring current via external shunt (for 4RU1SH2S)

The command for reading of the voltage over shunt is:

> snmpget -v1 -c public 192.168.1.100 1.3.6.1.4.1.19865.2.3.1.29.6.0

The received value is converted to Ampers with the help of the formula:

©2008-2024 www.ipnetcontrol.net

Iacc[Adc] = (SHmA/SHmv)*(3300*SNMPValue/1023)-1650)/22000,

where 'SHmA' is the shunt nominal current in milliampers, 'SHmV' is the voltage nominal – 60 or 75mV.

4.1.8 Access to the old OIDs for access to the I/O ports

For reverse compatibility in ***NetControl*** is retained the access to the objects that are supported by ***PicoIP*** (and older versions of ***NetControl***).

The main object is PicoIP.PortCTRL (1.3.6.1.4.1.19865.1.2.X.X.0) that supports group and individual access to the channels of the device. The concept, however, there is different – the channels are combined in bit masks and three groups. For more information you can go to the MIB file for the ***PicoIP*** (accessible on http://lan.neomontana-bg.com)

If you use the old objects you will not be able to use the 'Impulse' modes of the outputs and the option 'Invert Output'.

4.2. Sample PERL script for calculating the temperature, Unet and the alarm input

This sample PERL script uses the object for reading in Hex for all input/output channels (you must have installed the 'net-snmp' package). Only the necessary values for the temperature, voltage and alarm input are extracted and after a conversion are shown on the console in 1 second:

```
#!/usr/bin/perl
# #####
# PERL script testing the NetControl measurment inputs
#
#
# Created by Yasen Angelov, Neomontana Electronics, 2014
# Feel free to copy and modify
# #####

$host_ip = $ARGV[0];

if ($host_ip == '') {
	printf ("Usage: ./demo_netcontrol4.pl host_ip\n");
	printf ("Using defaul IP=192.168.1.100\n");
	#exit();
	$host_ip="192.168.1.100";
}

$community = "public";

printf("Host: %s\nTime		\tVIN		\tTemperature	\tAlarm\n", $host_ip);
printf("=============================================================================\n");
while (1)
{
	# Retreive all IO port values with single SNMP request (the iomReadAllHex.0 object)
	$all_ports=`snmpget -v1 -t 1 -r 1 -Oqv -c $community $host_ip .1.3.6.1.4.1.19865.2.3.2.2.0`;

	# Convert HEX string to array ($words[])containing two byte value of every channel
	my @hexs = split(' ', $all_ports);
	my @words = ();
	for ($k=0;$k<$#hexs;$k +=2) {
		#printf ("%d = %d.%d\n", $k/2, hex($hexs[$k]), hex($hexs[$k+1]) );
		$words[$k/2] = 256*hex($hexs[$k]) + hex($hexs[$k+1]);
	}

	# Get needed channels for voltage, temperature, alarm
	$snmpV = $words[25];
	$snmpT = $words[24];
	$snmpA = $words[30];

	# Convert ADC value to physical quantity
	$VINac=3.3*($snmpV/1023)*37.25 + 0.4;
	$VINdc=3.3*($snmpV/1023)*49 + 0.4;
	$Temperature=(3300*($snmpT/1023) - 500)/10.0;
	if ($snmpA > 0.75*1023) {
	 $Alarm="Open";
	}
	else {
	  $Alarm="Closed";
	}

	# Print line with date, values
	$tm = `date +%k:%M:%S`;
	chomp($tm);
	printf("%s ", $tm);

	printf("\t%5.1fVAC/%5.1fVdc[%d] \t%.1fC [%d]\t%s[%d]\n", $VINac, $VINdc, $snmpV, $Temperature, $snmpT, $Alarm, $snmpA);
	sleep(1);
}

 >[uesr@host]$ ./demo_netcontrol4.pl
Usage: ./demo_netcontrol4.pl host_ip
Using defaul IP=192.168.1.100
Host: 192.168.1.100
Time		VIN		Temperature	Alarm
=============================================================
15:47:33	2.8VAC/	3.6Vdc[20]	19.4C [215]	Open[1021]
15:47:34	2.8VAC/	3.6Vdc[20]	19.4C [215]	Open[1021]
```

5. Control via MQTT

5.1. Principle of operation of the MQTT protocol

The protocol uses the so-called topics or, in other words, objects that can be changed or read. For example, in ***NetControl*** we use the object "NetControl/in/ch9" to change the state of a relay channel 1 in models with 2, 4 or 8 relays; "NetControl/out/ ch9" is used when the state of the relay has changed (for example, if the status of the relay is changed via WEB Interface or SNMP)

After connecting to the server, the client subscribes to the objects of interest and thus tells the server to forward all messages related to those objects. In ***NetControl*** theese are all relay channels, so a SUBSCRIBE message is sent to server for each such object. Other clients connected to the server send a PUBLISH message to a topic/object and the server sends it to subscribed devices, so the ***NetControl*** relay can be switched by submitting a command to the server from another client (Domoticz, OpenHab, or other MQTT compatible communication software)

NetControl itself sends a PUBLISH message every time the status of the outputs is changed, as well as periodically for for sensor inputs.

A very accessible explanation of the principles of the MQTT protocol can be found at the following link:

https://www.hivemq.com/blog/mqtt-essentials/

5.2. MQTT Settings in NetControl

From 'IP Settings' menu you must first select „MQTT Server”, confirm the choice with 'Apply Settings' button and then enter the page with MQTT 'Settings'.

Server IP address	IP address of MQTT server
Server Port	TCP server port; defaylt port for MQTT service is 1883. Other values over 1024 are also allowed. *The MQTT standard also uses port 8883 for "secure" communication, which is **NOT** supported by NetControl.*
User name User password	User name and password for MQTT connection. Up to 32 characters.
MQTT QoS	Sets the QoS (Quality of Service) level of the communication to the server. **QoS2 mode is not supported!** QoS0 - (Receive data "at most once") - communication with the server is not confirmed within the MQTT protocol. Theoretically, data loss is possible, but practically unlikely due to underlying TCP. QoS1 - (Receive data “at least once”) - Receipt of each PUBLISH packet is confirmed by the other party. Duplicate messages may be possible.
Retain Flag in PUBLISH	Each time data is published to the server, this flag can be "fired", which tells the server to remember the last state of the object. Other clients, by connecting to the server and subscribing to data from the object in question, will receive its last status automatically from the server.
PUBLISH value format	Two formats are supported: JSON and RAW. The RAW format only sends an integer value each input/output channel

	(relay channel state is 1 or 0, ADC value for sensor inputs 0 ... 1023, etc.). The JSON format is the following: {"device": "MyNetControl", "value": {"real": "0%", "raw": 1}, "channel": 28, "type": 40;
Mirror /in to /out	All incoming commands in „/in“ chain will be forwarded to „/out” one.
User defined sub-topic	Allows insertation of additional "topic" into the device path. It may be empty. It is convenient for extra logical separation of multiple devices to a single server.
KeepAlive Period	Sets the time in seconds between sending a special Keep Alive message in MQTT connection, which is used to monitor the link. It is also possible to set a value of 0 (deactivation of this mechanism, which is acceptable by the MQTT protocol), but this may cause side effects in the TCP protocol and drop the connection if there is no other information transmitted.
Inputs auto-send period	All NetControl inputs (sensors, alarm, voltage, current, etc.) with this parameter can determine how often to send information about their current value to the MQTT server. This function can be deactivated with a value of 0. This periodic sending of information can replace the Keep-Alive mechanism. The Send Now button lets you manually send sensor data, which is convenient for diagnostics.

5.3. Topics/objects in MQTT for *NetControl* access

Outgoing from ***NetControl*** events (sensor data, altered relay state over Web or SNMP) use an object with the following format:

NetControl/subgroup/out/chXX

where:

'**NetControl**/' is a default prefix (factory fixed)

'**subgroup**/' is loaded from the „User defined sub-topic“ setting and may be completely absent, and then the format becomes 'NetControl/out/chXX'

'**out**/' default text that indicates that the event is "out" from the device (for example, 'Relay 1 turned on' or “Sensor 1 has value xxx”)

'**chXX**' is the NetControl channel where XX is its number (1, 2 ... 32) and corresponds to the channel numbers [P] in the SNMP description. „Virtual IO” channels are located at 'ch33' to 'ch40'.

Commands to a ***NetControl*** device must use a similar object, only the 'out/' portion being replaced by 'in/' - ie. we indicate that this is a command that is incoming to the device (for example, 'Turn on Relay 1').

NetControl uses another object (Will Topic) that is only output for the device:

NetControl/subgroup/out/status

When ***NetControl*** connects to the server, a PUBLISH is always sent to this object with the value "online", and the server is instructed when the link is lost to generate PUBLISH to the same topic with value 'offline'. This mechanism allows user to monitor whether the

device is connected to the server or not. A detailed explanation of the functioning of this mechanism can be found in the following link:

https://www.hivemq.com/blog/mqtt-essentials-part-9-last-will-and-testament

5.4. Data format

As it was clear in the previous section, two data formats are supported: RAW or JSON.

Commands sent towards ***NetControl*** use data only in RAW format: value is an integer such as 0 = switch off, 1 = on (text versions ON and OFF in uppercase only are also supported). For example (examples use the client mosquitto that is available for Linux, Windows, MAC):

Turn on relay at channel 9 (Line 1):

```
> mosquitto_pub -h 192.168.1.102 -t 'NetControl/subgroup/in/ch9' -m 1
```

Turn off relay at channel 9 (Line 1):

```
> mosquitto_pub -h 192.168.1.102 -t 'NetControl/subgroup/in/ch9' -m 0
```

With a similar command you can start or stop a macro:

```
> mosquitto_pub -h 192.168.1.102 -t 'NetControl/subgroup/in/macro1' -m 1
```

, with valid values being 1 or START and 0 or STOP.

Publishing (PUBLISH) data from NetControl can be in one of two formats (depending on which setting you have chosen). In the RAW format, the data is similar to SNMP access and the description in the SNMP chapters applies here (outputs return 0 or 1, analog/sensor inputs value 0..1023).

The JSON data format is much more convenient, mainly due to better compatibility with external automation clients (Node-RED, Domoticz, OpenHab etc.). For example, the alarm input data looks like this:

```
{"device":"MyNetControl",
„name”:”Channel X”,
"value":{"real":"OPEN","raw":1},
"channel":32,
"type":38,
"source":"auto"}
```

, where:

"**device**" is the host name you entered in the „I/O Settings menu“

„name” is the channels name (entered in „I/O Settings menu“)

"**value**" is the value of the channel and has two variants: real and raw. The advantage of real is that it reflects the user-set sensor (TDS300, HDS300, etc.) and is degrees, percentages, and so on. This allows you to directly use the value without further converting.Virtual IO channels use only the „real” field.

"**channel**" is the channel number (actually duplicates the number in the topic)

"**type**" is the channel type (TDS300, HDS300, etc.). For a list of channel types, see section 7.

"**source**" shows the data source: 'auto' is when the data is due to the periodic sending of sensor data; 'event' is when data is due to an external command (for example, on relay outputs) or “Macro”.

©2008-2024 www.ipnetcontrol.net

6. WiFi module

Some models are equipped with an optional 2.4GHz WiFi module operating at 802.11 b/g/n standards. For better signal strength, an external (3dBi) antenna of the RP-SMA connector is used.

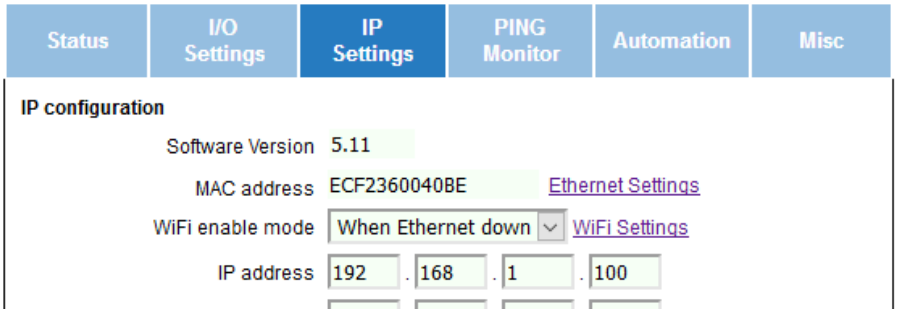
The device can only operate at one time with one of its communication interfaces: Ethernet or WiFi. By default, the device is in WiFi enabled mode when Ethernet connection is lost (physical loss of connection - ie disconnected cable). The IP Settings network settings are common to both connection modes.

WiFi and Ethernet share the same MAC address. That is why when switching between WiFi/Ethernet, the connection to the device may not appear immediately due to the transfer of the MAC address of the device between different ports of switches/routers.

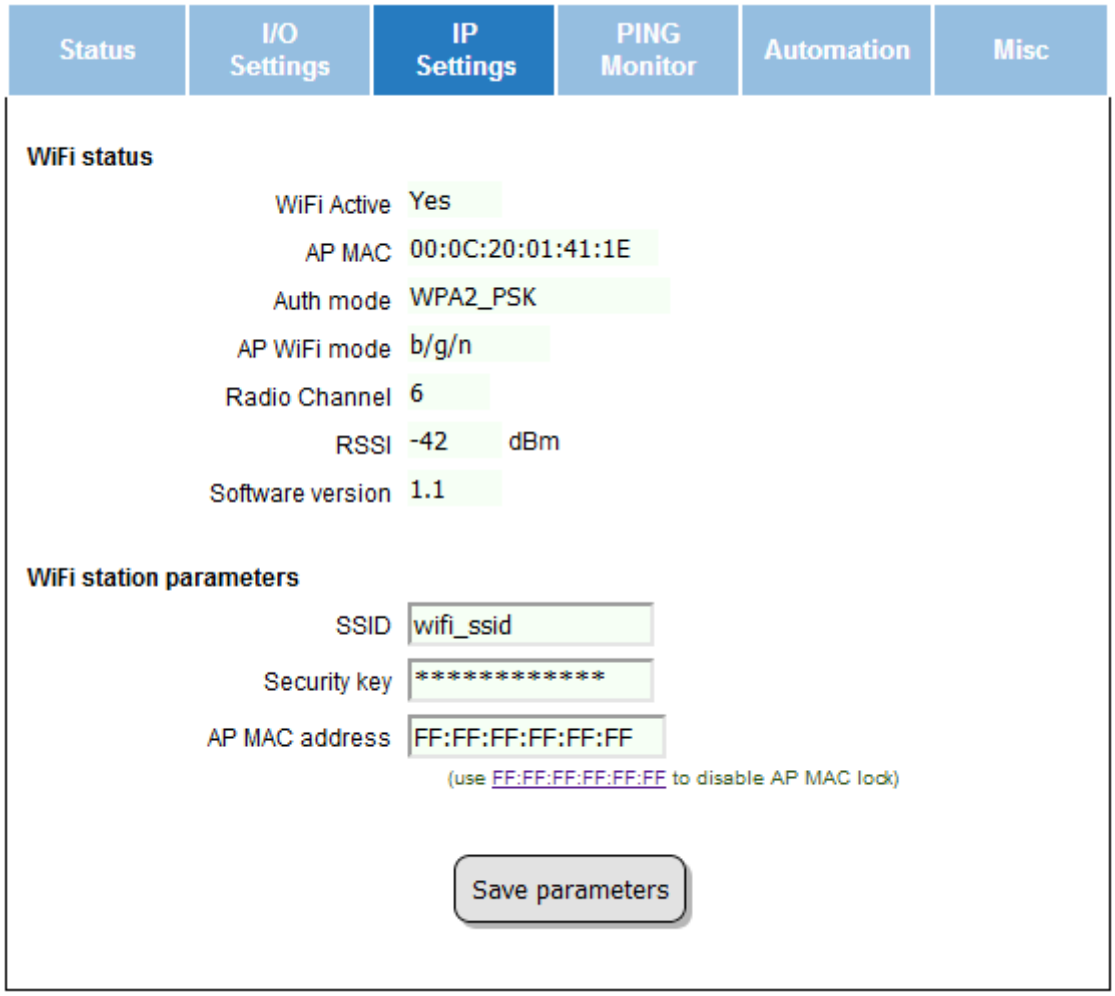
6.1.1 WiFi settings in the WEB interface

To establish WiFi connectivity, you only need to set SSID and SecurityKey (the other parameters, such as security mode, etc. are automatically selected).

The first setting is in the "IP Settings" menu - "WiFi Enable Mode". The possible modes are two "When Ethernet Down" (WiFi enabled when no Ethernet port is connected; default) and "Disabled" (fully deactivated WiFi).



The “WiFi Settings” link opens the Status and Settings page for the WiFi interface:



WiFi Active	Indicates whether the WiFi interface is currently active. When active, connection information is loaded in the following fields. You may need to reload the page if the data is not actual.
AP MAC	The MAC address of the access point to which the device is connected.
Auth mode	Current Connection Security Mode (Open, WEP, WPA_PSK, WPA2_PSK, WPA_WPA2_PSK)
AP WiFi mode	Supported communication / speed modes from hotspot. The highest parameter mode is selected.
Radio Channel	The radio channel on which the connection is established
RSSI	Signal strength indicator (greater value -> stronger signal)
Software version	Software version of the WiFi module

In the "WiFi station parameters" group you need to set the necessary parameters for making the connection:

SSID	The name of the wireless network you want to connect to. The factory default value for this field is: **wifi_ssid**
Security key	Password (key) to connect to this network. (asterisks do not match the real number of characters). If you only want to change some of the other

	parameters, do not fill in anything in this field and the current password will be kept. The factory default value for this field is: **wifi_secret**
AP MAC address	Setting a value other than FF:FF:FF:FF:FF:FF allows only access to an access point with the specified MAC address (and corresponding SSID).

802.1Q (VLAN) mode does not work in WiFi mode. Using this setting on WiFi will cause the device to lose connection.

The WiFi module supports firmware updates via TFTP (there is a "Start Firmware Update of WiFi module via TFTP" button in the "Misc" menu). The update file must be located in the root directory of the TFTP server and named "pipfw5_wifi.bin". For more information on TFTP updates, see Section 3.8.3. During the start of the TFTP update process, the WiFi module blocks switching between WiFi and Ethernet - the update process must start and end on the same communication channel.

©2008-2024 www.ipnetcontrol.net

7. APPENDIX I. Channel type in MQTT JSON data, ioModeXX.0 in SNMP and in ioreg.js 'PM' array (values in HEX)

Nubmer	Type/description
0	Manual Output
1	Toggle Output
2	Impulse Output
16	Phase regulator output
17	Zero-cross input for phase regulator
18	Pulse counter input
19	Analog Output 0..100%
32	Voltage (0-3.3Vdc)
33	Temperature TDS300
34	Temperature LM35Z
35	Voltage (0-33Vdc)
36	Voltage (0-110Vac)
37	Voltage (0-160Vdc)
38	Contact (Alarm) or VDS300, WDS300
39	External 75(60)mV shunt
40	Humidity HDS300
41	Voltage (0-62Vdc)
42	User Defined 1 (see 3.3.3)
43	User Defined 2 (see 3.3.3)
44	Voltage (0-10Vdc)
45	Voltage (0-22Vdc)
256	'Virtual IO' channel

©2008-2024 www.ipnetcontrol.net

8. APPENDIX II. Short user's manual for SNMP. List with the available objects in *NetControl*

The SNMP protocol defines different objects in every device which could be read or recorded according to their type. These objects has the so called OID – (object id) which characterize them explicitly. The said objects are described with a special syntax in a text file (MIB file) that allows the use of names instead of the difficult to remember digital equivalents of the OIDs.

The OIDs have tree like structure from the type .1.2.3.4.5.....and that way it is formed a unique number for every object. This structure is described in the respective MIB file. Neomontana Electronics has a registered 'branch' in this structure which is **1.3.6.1.4.1.19865.**

For access to the parameters of ***NetControl*** are used the commands snmpget and snmpset. The syntax is analogical.

>snmpget -v1 -c <парола read-only> <IP> <OID>

>snmpset -v1 -c <парола read-write> <IP> <OID> <тип данни> <стойност>

The command snmpset requires precise indication of the type of the data which will be sent to the respective OID. The permissible types are:

i: INTEGER, u: unsigned INTEGER, t: TIMETICKS, a: IPADDRESS
o: OBJID, s: STRING, x: HEX STRING, d: DECIMAL STRING, b: BITS
U: unsigned int64, I: signed int64, F: float, D: double

For reading the IP address is used:

> snmpget -v1 -c public 192.168.1.100 netIP.0

If there is no MIB file or poorly configured SNMP client the command above will not work and it will show an error. In this case either the problem with the MIB file should be resolved or to be used the digital equivalent (this option is universal and it always works with any SNMP client):

>snmpget -v1 -c public 192.168.1.100 .**1.3.6.1.4.1.19865.2.2.1.0**

In 'translation' to the digital equivalent it means:

(1.3.6.1.4.1.19865)	.2	.2	.1	.0
Neomontana	.*NetControl*	.global	. netIP	. 0

Other sample commands:

- read the state of output 'Line1'
  > snmpget -v1 -c public 192.168.1.100 ioValue9.0 i 1
  or
  > snmpget -v1 -c public 192.168.1.100 .1.3.6.1.4.1.19865.2.3.1.9.6.0
- setting of the level of output 'Line2'
  > snmpset -v1 -c private 192.168.1.100 ioValue10.0 i 1
  or
  > snmpset -v1 -c private 192.168.1.100 .1.3.6.1.4.1.19865.2.3.1.10.6.0 i 1
- setting of new IP address

  > snmpset -v1 -c private 192.168.1.100 netIP.0 a 192.168.1.145

- restart of device: snmpget -v1 -c public 192.168.1.100 .1.3.6.1.4.1.19865.2.1.3.0

Below is shown the result of the 'snmptranslate' command which composes the tree like structure with the names and the numbers of the objects and the sub objects. The digital equivalent could be easily extracted from it as you pass over the branches of the tree. From the tree it is obvious the types of the data which will be required when recording each object and whether it is read-only or read/write.

The repeating groups iop2...iop31, ipMon2...ipMon7, auto2...auto7 are hidden in order to shorten the list but the access to them in enabled.

©2008-2024 www.ipnetcontrol.net

```
>snmptranslate -Tp -IR -Ov netControl
+--Neomontana(19865)
+--netControl(2)
  |
  +--global(1)
  | |
  | +-- -R-- String    gloHostName(1)
  | |       Size: 0..14
  | +-- -R-- INTEGER   gloFWversion(2)
  | |       Range: 0..65535
  | +-- -R-- Null      gloReboot(3)
  | +-- -R-- Null      gloStartFirmwareUpdate(4)
  | +-- -RW- INTEGER   gloNetworkDefault(5)
  | |       Range: 0..65535
  | +-- -R-- INTEGER   gloMOD(127)
  |         Range: 0..65535
  |
  +--network(2)
  | |
  | +-- -RW- IpAddr    netIP(1)
  | +-- -R-- String    netMAC(2)
  | |       Textual Convention: PhysAddress
  | |       Size: 6
  | +-- -RW- INTEGER   netVLANTag(3)
  | |       Range: 0..4095
  | +-- -RW- String    netRWcommunity(4)
  | |       Size: 4..13
  | +-- -RW- String    netMACLock1(5)
  | |       Textual Convention: PhysAddress
  | |       Size: 6
  | +-- -RW- String    netMACLock2(6)
  | |       Textual Convention: PhysAddress
  | |       Size: 6
  | |
  | +--netReadAll(7)
  | |
  | +-- -RW- IpAddr    netDefGW(14)
  | +-- -RW- IpAddr    netNetMask(15)
  | +-- -RW- String    netROcommunity(17)
  | |       Size: 4..13
  | +-- -RW- IpAddr    netTrapServerIP(18)
  | +-- -RW- String    netTrapPassword(19)
  | |       Size: 4..13
  | +-- -RW- IpAddr    netAccessIP(20)
  | +-- -RW- IpAddr    netAccessMask(21)
  | +-- -RW- INTEGER   netHTTPport(22)
  | |       Range: 0..65535
  | +-- -RW- INTEGER   netSNMPport(23)
  | |       Range: 0..65535
  | +-- -RW- IpAddr    netTFTPServerIP(32)
  | +-- -RW- EnumVal   netBroadcastMode(64)
  | |       Values: Parse(0), Drop(1)
  | +-- -RW- EnumVal   net8021qMode(65)
  | |       Values: No(0), Yes(1)
  | +-- -RW- EnumVal   netTFTPupdateMode(66)
  | |       Values: No(0), Yes(1)
  | +-- -RW- EnumVal   netDHCPMode(67)
  | |       Values: No(0), Yes(1)
  | +-- -RW- EnumVal   netWebServerMode(68)
  | |       Values: No(0), Yes(1)
  | +-- -RW- EnumVal   netSPCMode(69)
  | |       Values: No(0), Yes(1)
  | +-- -RW- EnumVal   netSNMPNetConfigMode(70)
  |         Values: No(0), Yes(1)
  |
  +--io(3)
  | |
  | +--ioPorts(1)
  | | |
  | | +--iop1(1)
  | | | |
  | | | +-- -R-- INTEGER   ioNum1(1)
  | | | |       Range: 0..255
  | | | +-- -R-- String    ioName1(2)
  | | | |       Size: 0..14
  | | | +-- -RW- INTEGER   ioMode1(3)
  | | | |       Range: 0..255
  | | | +-- -RW- INTEGER   ioDefault1(4)
  | | | |       Range: 0..1
  | | | +-- -RW- INTEGER   ioImpulseFilter1(5)
  | | | |       Range: 0..65535
  | | | +-- -RW- INTEGER   ioValue1(6)
  | | | |       Range: 0..65535
  | | | +-- -R-- String    ioReadAll1(7)
  | | | |       Size: 0..255
  | | | +-- -R-- INTEGER    ioInvert1 (8)
  | | | |         Range: 0..1
  | | | +-- -R-- Gauge32    ioGauge1 (9)
  | | | |         Range: 0..4294967296
  | | | +-- -R-- String    ioPulseCfg1  (10)
  | | | |         Size: 0..255
  | | | |
..... iop2 to iop32skipped for shortness!
  | | |
  | | |
  | +--ioMisc(2)
  |   |
  |   +-- -R-- String    iomReadAllOld(1)
  |   |       Size: 1..80
  |   +-- -R-- String    iomReadAllHex(2)
  |   |       Size: 1..80
  |   +-- -RW- EnumVal   iomSetAll(3)
  |   |       Values: Off(0), On(1)
  |   +-- -RW- EnumVal   iomSaveIOMode(16)
  |           Values: No(0), Yes(1)
  |
  +--auto(4)
  | |
  | +--auto1(1)
  | | |
  | | +-- -RW- EnumVal   autMode1(1)
  | | |       Values: Disabled(0), Low(1), High(2), LowHigh(3), Acc(4)
  | | +-- -RW- INTEGER   autSourceIO1(2)
  | | |       Range: 0..255
  | | +-- -RW- INTEGER   autThresholdLow1(3)
  | | |       Range: 0..1023
  | | +-- -RW- INTEGER   autThresholdHigh1(4)
  | | |       Range: 0..1023
  | | +-- -RW- INTEGER   autTargetIO1(5)
  | | |       Range: 0..255
  | | +-- -RW- EnumVal   autTargetAction1(6)
  | | |       Values: On(0), Off(1)
  | | +-- -R-- String    autReadAll1(7)
  | |         Size: 0..80
  | |
  | | ... auto2 to auto7 skipped for shortness!
  | |
  | +--auto8(8)
  |   |
  |   +-- -RW- EnumVal   autMode8(1)
  |   |       Values: Disabled(0), Low(1), High(2), LowHigh(3), Acc(4)
  |   +-- -RW- INTEGER   autSourceIO8(2)
  |   |       Range: 0..255
  |   +-- -RW- INTEGER   autThresholdLow8(3)
  |   |       Range: 0..1023
  |   +-- -RW- INTEGER   autThresholdHigh8(4)
  |   |       Range: 0..1023
  |   +-- -RW- INTEGER   autTargetIO8(5)
  |   |       Range: 0..255
  |   +-- -RW- EnumVal   autTargetAction8(6)
  |   |       Values: On(0), Off(1)
  |   +-- -R-- String    autReadAll8(7)
  |           Size: 0..80
  |
  +--ipMon(5)
    |
    +--ipMon1(1)
    | |
    | +-- -RW- INTEGER   ipMonTimeout1(1)
    | |       Range: 0..65535
    | +-- -RW- IpAddr    ipMonTargetIP1(2)
    | +-- -RW- INTEGER   ipMonTargetIO1(3)
    | |       Range: 0..255
    | +-- -RW- EnumVal   ipMonTargetAction1(4)
    | |       Values: On(0), Off(1)
    | +-- -RW- INTEGER   ipMonEventsLimit1(5)
    | |       Range: 0..255
    | +-- -R-- INTEGER   ipMonTimeLeft1(6)
    | |       Range: 0..255
    | +-- -R-- String    ipMonReadAll1(7)
    |         Size: 0..80
    | +-- -RW-INTEGER   ipMonPingSize1(8)
    |         Range: 32..1472
    |
    | ... ipmon2 to ipmon7 skipped for shortness!
    |
    +--ipMon8(8)
      |
      +-- -RW- INTEGER   ipMonTimeout8(1)
      |       Range: 0..65535
      +-- -RW- IpAddr    ipMonTargetIP8(2)
      +-- -RW- INTEGER   ipMonTargetIO8(3)
      |       Range: 0..255
      +-- -RW- EnumVal   ipMonTargetAction8(4)
      |       Values: On(0), Off(1)
      +-- -RW- INTEGER   ipMonEventsLimit8(5)
      |       Range: 0..255
      +-- -R-- INTEGER   ipMonTimeLeft8(6)
      |       Range: 0..255
      +-- -R-- String    ipMonReadAll8(7)
      |       Size: 0..80
      +-- -RW-INTEGER   ipMonPingSize8(8)
              Range: 32..1472
```