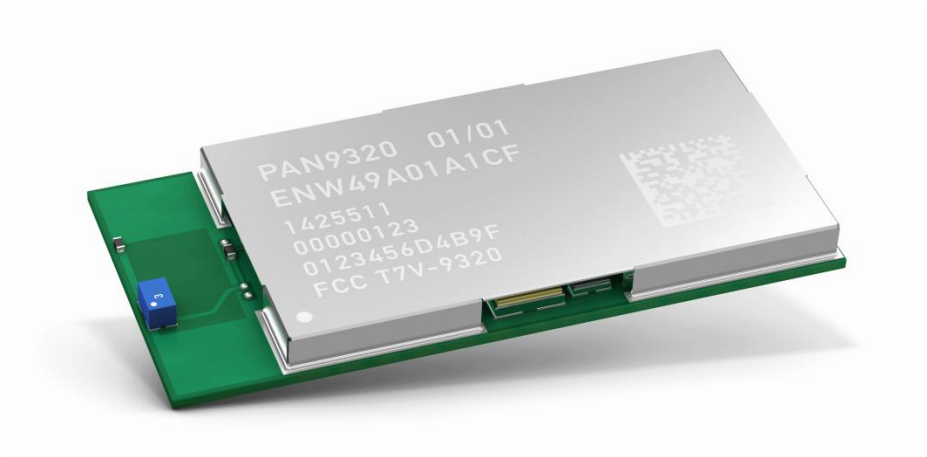


# PAN9320

Firmware Version 1.9.0.1

## Command Specification

Rev. 1.3



By purchase of any of the products described in this document the customer accepts the document's validity and declares their agreement and understanding of its contents and recommendations. Panasonic Industrial Devices Europe GmbH (Panasonic) reserves the right to make changes as required at any time without notification.

© Panasonic Industrial Devices Europe GmbH 2017.

All rights reserved.

This Command Specification does not lodge the claim to be complete and free of mistakes.

# Table of Contents

<b>1</b>	<b>About This Document</b> .....	<b>4</b>
1.1	Purpose and Audience .....	4
1.2	Revision History.....	4
1.3	Use of Symbols .....	5
1.4	Related Documents .....	5
<b>2</b>	<b>Overview</b> .....	<b>6</b>
<b>3</b>	<b>Commands</b> .....	<b>7</b>
3.1	WLAN Service (Module: wlan).....	9
3.2	Name Services (Module: name) .....	20
3.3	NET Service (Module: net) .....	25
3.4	Email (Module: smail) .....	29
3.5	System (Module: system) .....	33
3.6	User Management (Module: user) .....	38
3.7	Telnet (Module: telnet).....	41
3.8	Firmware Update (Module: fwu) .....	42
3.9	Command UART (Module: cmduart) .....	44
3.10	GPIO (Module: gpio).....	47
3.11	HTTP Client (Module: httpc) .....	49
3.12	Netcat (Module: netcat) .....	52
3.13	UART Binary (Module: binuart).....	59
3.14	Time (Module: time).....	61
<b>4</b>	<b>Communication with the Host Controller</b> .....	<b>63</b>
4.1	Command UART .....	63
4.2	Binary UART.....	63
<b>5</b>	<b>Status Information</b> .....	<b>64</b>
5.1	Telegram Return Code .....	64
5.2	Wi-Fi Parameter .....	65
5.3	Mail Service .....	65
5.4	User Management .....	66
5.5	Firmware Update .....	67
5.6	HTTP Client .....	67
5.7	UART Configuration.....	67
5.8	GPIO.....	68
<b>6</b>	<b>Contact Details</b> .....	<b>69</b>
6.1	Contact Us.....	69
6.2	Product Information .....	69

# 1 About This Document




## 1.1 Purpose and Audience

This Command Specification provides details on the communication via UART and HTTP/JSON for the various features of the PAN9320 module. It is intended for software engineers. The product is referred to as “the PAN9320” or “the module” within this document.

## 1.2 Revision History

Revision	Date	Revised by	Modifications/Remarks
1.01	22.09.2014	bme	1st preliminary version
1.02	22.01.2015	osu	Entire chapter 2 revised
1.03	26.01.2015	bme, ch	Commands adapted
1.04	19.02.2015	bme, ch	Netcat added
1.05	23.02.2015	ch	Range of the Parameter 1 “baudrate” in netcat uart_cfg added
1.06	10.3.2015	ch	Layout changed
1.07	20.3.2015	ch	Modul information added
1.1	13.7.2015	ch	Minor changes
1.2	27.11.2015	ch	Ranges, detail corrected
1.3	26.6.2017	dhe	New document format and layout

## 1.3 Use of Symbols

Symbol	Description
	<p><b>Note</b></p> <p>Indicates important information for the proper use of the product. Non-observance can lead to errors.</p>
	<p><b>Attention</b></p> <p>Indicates important notes that, if not observed, can put the product's functionality at risk.</p>
	<p><b>Tip</b></p> <p>Indicates useful information designed to facilitate working with the software.</p>
⇒ [chapter number] [chapter title]	<p><b>Cross reference</b></p> <p>Indicates cross references within the document.</p> <p><b>Example:</b></p> <p>Description of the symbols used in this document ⇒ <a href="#">1.3 Use of Symbols</a>.</p>
This font	<p><b>File names, messages, user input</b></p> <p>Indicates file names or messages and information displayed on the screen or to be selected or entered by the user.</p> <p><b>Examples:</b></p> <p><code>pan1760.c</code> contains the actual module initialization.</p> <p>The message <code>Failed to save your data is displayed</code>.</p> <p>Enter the value <code>Product 123</code>.</p>

## 1.4 Related Documents

Please refer to the Panasonic website for related documents ⇒ [6.2 Product Information](#).

## 2 Overview

The PAN9320 is a 2.4 GHz 802.11 b/g/n embedded Wi-Fi module with integrated stack and API that minimizes firmware development and includes a full security suite. The module is specifically designed for highly integrated and cost-effective applications. The module includes a fully shielded case, integrated crystal oscillators, and a chip antenna.

The module combines a high-performance CPU, high sensitivity wireless radio, baseband processor, medium access controller, encryption unit, boot ROM with patching capability, internal SRAM, and in-system programmable flash memory. The module's integrated memory is available to the application for storing web content such as HTML pages or image data.

Parallel support of access point and infrastructure mode allows easy setup up of simultaneous Wi-Fi connections from the module to smart devices and home network routers.

The pre-programmed Wi-Fi SoC firmware enables client (STA) and micro access point ( $\mu$ AP) applications. With the transparent mode, raw data can be sent from the UART to the air interface to smart devices, web servers, or PC applications.

Please refer to the Panasonic website for related documents [⇒ 6.2 Product Information](#).

## 3 Commands



There are three types of commands:

- Request
- Response
- Error

The format is ASCII-based, ending with "CR-LF".

### „Request“ Command

- Request data
- Set values
- Trigger various functionalities

#### Structure

Command	Module	Variable	Parameter
---------	--------	----------	-----------

Definition	
<b>Command</b>	Either "get" or "set".
<b>Module</b>	Defines the functional software module which shall be addressed.
<b>Variable</b>	A subset of the selected module.
<b>Parameter</b>	Different parameters can be used depending on the module and variable.

Each request triggers a „Response“ message of the following structure:

### „Response“ Command

#### Structure

Command	Module	Variable	Return Code	Parameter
---------	--------	----------	-------------	-----------

Definition	
<b>Command</b>	Either "get" or "set".
<b>Module</b>	Defines the functional software module which shall be addressed.
<b>Variable</b>	A subset of the selected module.
<b>Return Code</b>	Codes ⇒ <a href="#">5.1 Telegram Return Code</a> .
<b>Parameter</b>	Different parameters can be used depending on the module and variable.

## „Error“ Message

### Structure

„Error“	Return Code
---------	-------------

Further information on the return code ⇒ [5.1 Telegram Return Code](#).

## Command Architecture

Examples	
HTTP/JSON Request	[ "Command", "Module", "Variable" ]
HTTP/JSON Response	[ "Command", "Module", "Variable", "Return-Code", "Parameter" ]
CMD-UART Request	Command Module Variable\x0d\x0a
CMD-UART Response	Command Module Variable Return-Code Parameter\x0d\x0a



### 3.1 WLAN Service (Module: wlan)

#### 3.1.1 Infrastructure SSID (Variable: infra\_ssid)

Definition	
<b>Command Option</b>	Get
<b>Parameter 1</b>	<p><b>&lt;ssid&gt;</b> SSID of the infrastructure network to which the application is connected.</p> <p><b>Default value</b> -</p> <p><b>Range</b> String with ASCII-printable characters with a length of (1-32).</p>
<b>Description</b>	Returns the SSID of the WLAN network to which the application is connected as an ASCII-encoded string with a maximum length of 32 characters.
<b>GET Rights</b>	0x00

Examples	
HTTP/JSON Get-Request	<code>["get", "wlan", "infra_ssid"]</code>
HTTP/JSON Get-Response	<code>["get", "wlan", "infra_ssid", "0", "testnetwork"]</code>
CMD-UART Get-Request	<code>get wlan infra_ssid\x0d\x0a</code>
CMD-UART Get-Response	<code>get wlan infra_ssid 0 testnetwork\x0d\x0a</code>

#### 3.1.2 Infrastructure Mode (Variable: infra\_mode)

Definition	
<b>Command Option</b>	Get/set
<b>Parameter 1</b>	<p><b>&lt;mode&gt;</b> Enables/disables the infrastructure mode.</p> <p><b>Default value</b> off</p> <p><b>Range</b> [on, off]  <b>on:</b> mode enabled  <b>off:</b> mode disabled</p>
<b>Description</b>	The variable returns the status of the infrastructure mode. It can also enable or disable the infrastructure mode.
<b>GET Rights</b>	0x00
<b>SET Rights</b>	0x02

Examples	
HTTP/JSON Get-Request	<code>["get", "wlan", "infra_mode"]</code>
HTTP/JSON Get-Response	<code>["get", "wlan", "infra_mode", "0", "on"]</code>
HTTP/JSON Set-Request	<code>["set", "wlan", "infra_mode", "off"]</code>
HTTP/JSON Set-Response	<code>["set", "wlan", "infra_mode", "0", "off"]</code>
CMD-UART Get-Request	<code>get wlan infra_mode\x0d\x0a</code>
CMD-UART Get-Response	<code>get wlan infra_mode 0 on\x0d\x0a</code>
CMD-UART Set-Request	<code>set wlan infra_mode off\x0d\x0a</code>
CMD-UART Set-Response	<code>set wlan infra_mode 0 off\x0d\x0a</code>

### 3.1.3 Infrastructure Status (Variable: infra\_status)

Definition	
<b>Command Option</b>	Get
<b>Parameter 1</b>	<p><b>&lt;status&gt;</b> Status of the infrastructure connection.</p> <p><b>Default value</b> 0</p> <p><b>Range</b> ⇒ <a href="#">Wi-Fi Status Information</a></p>
<b>Description</b>	Returns the status of the infrastructure connection ⇒ <a href="#">Wi-Fi Status Information</a> .
<b>GET Rights</b>	0x00

Examples	
HTTP/JSON Get-Request	<code>["get", "wlan", "infra_status"]</code>
HTTP/JSON Get-Response	<code>["get", "wlan", "infra_status", "0", "4"]</code>
CMD-UART Get-Request	<code>get wlan infra_status\x0d\x0a</code>
CMD-UART Get-Response	<code>get wlan infra_status 0 4\x0d\x0a</code>

### 3.1.4 Infrastructure Security (Variable: infra\_sec)

Definition	
<b>Command Option</b>	Get
<b>Parameter 1</b>	<p><b>&lt;security&gt;</b> Security setting of the infrastructure network.</p> <p><b>Default value</b> -</p> <p><b>Range</b> [1..5]</p> <p>1: OPEN 2: WEP 3: WPA 4: WPA2 5: WPA2_MIX</p>
<b>Description</b>	Returns the security setting of the infrastructure network to which the application is connected.
<b>GET Rights</b>	0x00

Examples	
HTTP/JSON Get-Request	<code>["get", "wlan", "infra_sec"]</code>
HTTP/JSON Get-Response	<code>["get", "wlan", "infra_sec", "0", "2"]</code>
CMD-UART Get-Request	<code>get wlan infra_sec\x0d\x0a</code>
CMD-UART Get-Response	<code>get wlan infra_sec 0 2\x0d\x0a</code>

### 3.1.5 Infrastructure Configuration (Variable: infra\_cfg)

Definition	
<b>Command Option</b>	Set
<b>Parameter 1</b>	<p><b>&lt;ssid&gt;</b> SSID of the selected infrastructure network.</p> <p><b>Default value</b> -</p> <p><b>Range</b> String with ASCII-printable characters with a length of (1-32).</p>
<b>Parameter 2</b>	<p><b>&lt;psk&gt;</b> Pass phrase of the chosen infrastructure network.</p> <p><b>Default value</b> -</p>
<b>Parameter 3</b>	<p><b>&lt;security&gt;</b> Security setting of the infrastructure network.</p> <p><b>Default value</b> -</p> <p><b>Range</b> [1..5]                      1: OPEN                      2: WEP                      3: WPA                      4: WPA2                      5: WPA2_MIX</p>
<b>Description</b>	A connection to a selected infrastructure network can be established with this variable. The pass phrase and the security level must match the settings of the infrastructure network.
<b>SET Rights</b>	0x02

Examples	
HTTP/JSON Set-Request	<code>["set", "wlan", "infra_cfg", "testnetwork", "password", "2"]</code>
HTTP/JSON Set-Response	<code>["set", "wlan", "infra_cfg", "0", "testnetwork", "password", "2"]</code>
CMD-UART Set-Request	<code>set wlan infra_cfg testnetwork password 2\x0d\x0a</code>
CMD-UART Set-Response	<code>set wlan infra_cfg 0 testnetwork password 2\x0d\x0a</code>

### 3.1.6 Access Point SSID (Variable: ap\_ssid)

Definition	
<b>Command Option</b>	Get
<b>Parameter 1</b>	<p><b>&lt;ssid&gt;</b> SSID of the access point.</p> <p><b>Default value</b> PAN9320_AP</p> <p><b>Default password:</b> PAN_9320</p> <p><b>Range</b> String with ASCII-printable characters with a length of (1-32).</p>
<b>Description</b>	Returns the SSID of the access point as an ASCII-encoded string with a maximum length of 32 characters.
<b>GET Rights</b>	0x00

Examples	
HTTP/JSON Get-Request	["get", "wlan", "ap_ssid"]
HTTP/JSON Get-Response	["get", "wlan", "ap_ssid", "0", "apwlannetz"]
CMD-UART Get-Request	get wlan ap_ssid\x0d\x0a
CMD-UART Get-Response	get wlan ap_ssid 0 apwlannetz\x0d\x0a

### 3.1.7 Access Point Mode (Variable: ap\_mode)

Definition	
<b>Command Option</b>	Get/set
<b>Parameter 1</b>	<p><b>&lt;mode&gt;</b> Enables/disables the access point mode.</p> <p><b>Default value</b> on</p> <p><b>Range</b> [on, off]</p> <p><b>on:</b> mode enabled</p> <p><b>off:</b> mode disabled</p>
<b>Description</b>	The variable returns the status of the access point mode. It can also enable or disable the access point mode.
<b>GET Rights</b>	0x00
<b>SET Rights</b>	0x04

Examples	
HTTP/JSON Get-Request	["get", "wlan", "ap_mode"]
HTTP/JSON Get-Response	["get", "wlan", "ap_mode", "0", "on"]
HTTP/JSON Set-Request	["set", "wlan", "ap_mode", "off"]
HTTP/JSON Set-Response	["set", "wlan", "ap_mode", "0", "off"]
CMD-UART Get-Request	get wlan ap_mode\x0d\x0a
CMD-UART Get-Response	get wlan ap_mode 0 on\x0d\x0a
CMD-UART Set-Request	set wlan ap_mode off\x0d\x0a
CMD-UART Set-Response	set wlan ap_mode 0 off\x0d\x0a

### 3.1.8 Access Point Security (Variable: ap\_sec)

Definition	
<b>Command Option</b>	Get
<b>Parameter 1</b>	<p><b>&lt;security&gt;</b> Security setting of the access point network.</p> <p><b>Default value</b> 4</p> <p><b>Range</b> [1..5]</p> <p>1: OPEN 2: WEP 3: WPA 4: WPA2 5: WPA2_MIX</p>
<b>Description</b>	Returns the security setting of the access point network.
<b>GET Rights</b>	0x00

Examples	
HTTP/JSON Get-Request	<code>["get", "wlan", "ap_sec"]</code>
HTTP/JSON Get-Response	<code>["get", "wlan", "ap_sec", "0", "2"]</code>
CMD-UART Get-Request	<code>get wlan ap_sec\x0d\x0a</code>
CMD-UART Get-Response	<code>get wlan ap_sec 0 2\x0d\x0a</code>

### 3.1.9 Access Point Configuration (Variable: variable: ap\_cfg)

Definition	
<b>Command Option</b>	Set
<b>Parameter 1</b>	<p><b>&lt;ssid&gt;</b> SSID of the access point.</p> <p><b>Default value</b> -</p> <p><b>Range</b> String with ASCII-printable characters with a length of (1-32).</p>
<b>Parameter 2</b>	<p><b>&lt;psk&gt;</b> Pass phrase of the access point.</p> <p><b>Default value</b> -</p> <p><b>Range</b> String with ASCII-printable characters with a length of (8-63).</p>
<b>Parameter 3</b>	<p><b>&lt;security&gt;</b> Security level of the access point.</p> <p><b>Default value</b> -</p> <p><b>Range</b> 1: OPEN 2: WEP /* Not supported */ 3: WPA 4: WPA2 5: WPA2_MIX</p>
<b>Parameter 4</b>	<p><b>&lt;radio channel&gt;</b> Radio channel for the AP.</p> <p>The radio channel is automatically adapted when the WLAN infrastructure interface connects to a WLAN.</p>
<b>Description</b>	Configures the access point.
<b>GET Rights</b>	0x04

Examples	
HTTP/JSON Set-Request	<code>["set", "wlan", "ap_cfg", "apwlan", "psk", "4", "0"]</code>
HTTP/JSON Set-Response	<code>["set", "wlan", "ap_cfg", "0", "apwlan", "psk", "4", "0"]</code>
CMD-UART Set-Request	<code>set wlan ap_cfg apwlan psk 4 0\x0d\x0a</code>
CMD-UART Set-Response	<code>set wlan ap_cfg 0 apwlan psk 4 0\x0d\x0a</code>

### 3.1.10 Access Point Station List (Variable: ap\_stlist)

Definition	
<b>Command Option</b>	Get
<b>Parameter 1</b>	<b>&lt;stlist&gt;</b> List of connected devices (MAC address).
<b>Description</b>	A list of connected devices can be requested with this parameter. The MAC address of the device is listed.
<b>GET Rights</b>	0x00

Examples	
HTTP/JSON Set-Request	<code>["get","wlan","ap_stlist"]</code>
HTTP/JSON Set-Response	<code>["get","wlan","ap_stlist","0","01:02:03:04:05:06", "11:12:13:14:15:16"]</code>
CMD-UART Set-Request	<code>get wlan ap_stlist\x0d\x0a</code>
CMD-UART Set-Response	<code>get wlan ap_stlist 0 01:02:03:04:05:06 11:12:13:14:15:16 \x0d\x0a</code>

### 3.1.11 Wireless Network Scan (Variable: scan)

Definition	
<b>Command Option</b>	Set
<b>Description</b>	Triggers a scan for available Wireless networks.
<b>SET Rights</b>	0x02

Examples	
HTTP/JSON Set-Request	<code>["set","wlan","scan"]</code>
HTTP/JSON Set-Response	<code>["set","wlan","scan","0"]</code>
CMD-UART Set-Request	<code>set wlan scan\x0d\x0a</code>
CMD-UART Set-Response	<code>set wlan scan 0\x0d\x0a</code>



### 3.1.12 Available Wireless Networks (Variable: list)

Definition	
<b>Command Option</b>	Get
<b>Parameter 1</b>	<p><b>&lt;ssid&gt;</b> SSID of the detected wireless network.</p> <p><b>Default value</b> -</p> <p><b>Range</b> String with ASCII-printable characters with a length of (1-32).</p>
<b>Parameter 2</b>	<p><b>&lt;security&gt;</b> Security level of the detected wireless network.</p> <p><b>Default value</b> -</p> <p><b>Range</b> [1..5]</p> <p>1: OPEN 2: WEP 3: WPA 4: WPA2 5: WPA2_MIX</p>
<b>Parameter 3</b>	<p><b>&lt;signal&gt;</b> Signal strength of the detected wireless network.</p> <p><b>Default value</b> -</p> <p><b>Range</b> 0-100</p>
<b>Description</b>	Returns a list of available wireless networks which were found by the network scan. Up to eight detected networks will be returned in a single Get response. Each detected network is described with its SSID, security type, and signal strength. The first possible three parameters (1-3) describe the first network in the list. The next possible three parameters (4-6) describe the second network in the list and so on, until the last possible (maximum) three parameters (16-18) for the eighth and last network in the list.
<b>GET Rights</b>	0x00

Examples	
HTTP/JSON Get-Request	<code>["get","wlan","list"]</code>
HTTP/JSON Get-Response	<code>["get","wlan","list","0","net1","2","50","net2","1","80"]</code>
CMD-UART Get-Request	<code>get wlan list\x0d\x0a</code>
CMD-UART Get-Response	<code>get wlan list 0 net1 2 50 net2 1 80\x0d\x0a</code>

### 3.1.13 WLAN Region Code (Variable: rcode)

Definition	
<b>Command Option</b>	Get
<b>Parameter 1</b>	<b>&lt;rcode&gt;</b> Region code <b>Default value</b> EU <b>Range</b> [EU, US]
<b>Description</b>	Region code for the radio chip.
<b>GET Rights</b>	0x00

Examples	
HTTP/JSON Get-Request	<code>["get", "wlan", "rcode"]</code>
HTTP/JSON Get-Response	<code>["get", "wlan", "rcode", "0", "EU"]</code>
CMD-UART Get-Request	<code>get wlan rcode\x0d\x0a</code>
CMD-UART Get-Response	<code>get wlan rcode 0 EU\x0d\x0a</code>

### 3.1.14 WLAN BSSID (Variable: infra\_bssid)

Definition	
<b>Command Option</b>	Get
<b>Parameter 1</b>	<b>&lt;bssid&gt;</b> Current BSSID <b>Default value</b> -
<b>Description</b>	BSSID of the connected Wi-Fi.
<b>GET Rights</b>	0x00

Examples	
HTTP/JSON Get-Request	<code>["get", "wlan", "infra_bssid"]</code>
HTTP/JSON Get-Response	<code>["get", "wlan", "infra_bssid", "0", "80:70:60:50:40:30"]</code>
CMD-UART Get-Request	<code>get wlan infra_bssid\x0d\x0a</code>
CMD-UART Get-Response	<code>get wlan infra_bssid 0 80:70:60:50:40:30\x0d\x0a</code>

### 3.1.15 WLAN Channel (Variable: infra\_rch)

Definition	
<b>Command Option</b>	Get
<b>Parameter 1</b>	<b>&lt;channel&gt;</b> Current radio channel <b>Default value</b> -
<b>Description</b>	Radio channel of the connected Wi-Fi.
<b>GET Rights</b>	0x00

Examples	
HTTP/JSON Get-Request	["get","wlan","infra_rch"]
HTTP/JSON Get-Response	["get","wlan","infra_rch","0","5"]
CMD-UART Get-Request	get wlan infra_rch\x0d\x0a
CMD-UART Get-Response	get wlan infra_bss rch id 0 5\x0d\x0a

### 3.1.16 WLAN RSSI (Variable: infra\_rssi)

Definition	
<b>Command Option</b>	Get
<b>Parameter 1</b>	<rssi>                      Current RSSI <b>Default value</b> -
<b>Description</b>	RSSI of the connected Wi-Fi.
<b>GET Rights</b>	0x00

Examples	
HTTP/JSON Get-Request	["get","wlan","infra_rssi"]
HTTP/JSON Get-Response	["get","wlan","infra_rssi","0","80"]
CMD-UART Get-Request	get wlan infra_rssi\x0d\x0a
CMD-UART Get-Response	get wlan infra_rssi rch id 0 80\x0d\x0a

### 3.2 Name Services (Module: name)

#### 3.2.1 Device Name (Variable: device)

Definition	
<b>Command Option</b>	Set
<b>Parameter 1</b>	<p><b>&lt;name&gt;</b> Device name</p> <p><b>Default value</b> -</p> <p><b>Range</b> String with a maximum of 15 characters.</p>
<b>Description</b>	<p>The device name is a universal name and it is used for the following services:</p> <ul style="list-style-type: none"> <li>• MDNS-Domain</li> <li>• MDNS-Serv1-Inst-Name</li> <li>• MDNS-Serv2-Inst-Name</li> <li>• NBNS-Name</li> </ul> <p>The device name is limited to twenty-six letters, ten digits, and the hyphen character. It is not allowed to use spaces or other punctuation.</p> <p>The Get command is not useful, because the set device name can differ, if one or more name services detect name conflicts. Nonetheless, it is possible to get and also set the service name separately from each name service</p> <p>⇒ <a href="#">3.2.4 mDNS Server 1/2 (Variable: mdns_serv1    mdns_serv2)</a> and</p> <p>⇒ <a href="#">3.2.5 NetBIOS Name Service Name (Variable: nbns_name)</a>.</p>
<b>SET Rights</b>	0x02

Examples	
HTTP/JSON Set-Request	<code>["set", "name", "device", "pan9320"]</code>
HTTP/JSON Set-Response	<code>["set", "name", "device", "0", "pan9320"]</code>
CMD-UART Set-Request	<code>set name device pan9320\x0d\x0a</code>
CMD-UART Set-Response	<code>set name device 0 pan9320\x0d\x0a</code>

#### 3.2.2 MDNS Domain (Variable: mdns\_domain)

Definition	
<b>Command Option</b>	Get/set
<b>Parameter 1</b>	<p><b>&lt;name&gt;</b> mDNS domain</p> <p><b>Default value</b> pan9320</p> <p><b>Range</b> Limited by the software with a maximum of 31 characters.</p>
<b>Description</b>	<p>The domain is for the mDNS service. The mDNS domain is limited to twenty-six letters, ten digits, and the hyphen character. It is not allowed to use spaces or other punctuation.</p>
<b>GET Rights</b>	0x00
<b>SET Rights</b>	0x04

Examples	
HTTP/JSON Get-Request	["get", "name", "mdns_domain"]
HTTP/JSON Get-Response	["get", "name", "mdns_domain", "0", "pan9320"]
HTTP/JSON Set-Request	["set", "name", "mdns_domain", "pan9320"]
HTTP/JSON Set-Response	["set", "name", "mdns_domain", "0", "pan9320"]
CMD-UART Get-Request	get name mdns_domain\x0d\x0a
CMD-UART Get-Response	get name mdns_domain 0 pan9320\x0d\x0a
CMD-UART Set-Request	set name mdns_domain pan9320\x0d\x0a
CMD-UART Set-Response	set name mdns_domain 0 pan9320\x0d\x0a

### 3.2.3 mDNS Mode (Variable: mdns\_mode)

Definition	
<b>Command Option</b>	Get/set
<b>Parameter 1</b>	<p><b>&lt;mode&gt;</b> Enables/Disables mDNS mode.</p> <p><b>Default value</b> on</p> <p><b>Range</b> [on, off]</p> <p><b>on:</b> mode enabled</p> <p><b>off:</b> mode disabled</p>
<b>Description</b>	The mDNS mode is used to switch the mDNS service on or off.
<b>GET Rights</b>	0x00
<b>SET Rights</b>	0x04

Examples	
HTTP/JSON Get-Request	["get", "name", "mdns_mode"]
HTTP/JSON Get-Response	["get", "name", "mdns_mode", "0", "on"]
HTTP/JSON Set-Request	["set", "name", "mdns_mode", "on"]
HTTP/JSON Set-Response	["set", "name", "mdns_mode", "0", "on"]
CMD-UART Get-Request	get name mdns_mode\x0d\x0a
CMD-UART Get-Response	get name mdns_mode 0 on\x0d\x0a
CMD-UART Set-Request	set name mdns_mode on\x0d\x0a
CMD-UART Set-Response	set name mdns_mode 0 on\x0d\x0a

### 3.2.4 mDNS Server 1/2 (Variable: mdns\_serv1 || mdns\_serv2)

Definition	
<b>Command Option</b>	Get/set
<b>Parameter 1</b>	<p><b>&lt;dev_name&gt;</b> Device name</p> <p><b>Default value</b> pan9320</p> <p><b>Range</b> String with a maximum of 31 characters.</p> <p>The device name is limited to twenty-six letters, ten digits, and the hyphen character. It is not allowed to use spaces or other punctuation.</p>
<b>Parameter 2</b>	<p><b>&lt;prot_name&gt;</b> Protocol name</p> <p><b>Default value</b> _http._tcp</p> <p><b>Range</b> String with a maximum of 31 characters.</p>
<b>Parameter 3</b>	<p><b>&lt;port&gt;</b> Port of the service</p> <p><b>Default value</b> 80</p> <p><b>Range</b> 0-65535</p>
<b>Parameter 4</b>	<p><b>&lt;tll&gt;</b> Time to live of the service.</p> <p><b>Default value</b> 120</p> <p><b>Range</b> 0-4294967295</p>
<b>Parameter 5</b>	<p><b>&lt;txt&gt;</b> Additional information of the service.</p> <p><b>Default value</b> pan9320</p> <p><b>Range</b> String with maximum of 31 characters</p>
<b>Description</b>	The mDNS service can be configured for individual use. When the <prot_name> is set, every device can propagate its own service in the network.
<b>GET Rights</b>	0x00
<b>SET Rights</b>	0x04

Examples	
HTTP/JSON Get-Request	["get", "name", "mdns_serv1"]
HTTP/JSON Get-Response	["get", "name", "mdns_serv1", "0", "pan9320", "_http._tcp", "80", "120", "PAN9320text"]
HTTP/JSON Set-Request	["set", "name", "mdns_serv1", "pan9320", "_http._tcp", "80", "120", "PAN9320text"]
HTTP/JSON Set-Response	["set", "name", "mdns_serv1", "0", "pan9320", "_http._tcp", "80", "120", "PAN9320text"]
CMD-UART Get-Request	get name mdns_serv1\x0d\x0a
CMD-UART Get-Response	get name mdns_serv1 0 pan9320 _http._top 80 120 PAN9320text\x0d\x0a
CMD-UART Set-Request	set name mdns_serv1 pan9320 _http._top 80 120 PAN9320text\x0d\x0a
CMD-UART Set-Response	set name mdns_serv1 0 pan9320 _http._top 80 120 PAN9320text\x0d\x0a

### 3.2.5 NetBIOS Name Service Name (Variable: nbns\_name)

Definition	
<b>Command Option</b>	Get/set
<b>Parameter 1</b>	<p><b>&lt;name&gt;</b> NetBIOS Name Service name.</p> <p><b>Default value</b> pan9320</p> <p><b>Range</b> String with a maximum size of 15 characters.</p>
<b>Description</b>	The NetBIOS Name Service Name is limited to 15 characters.
<b>GET Rights</b>	0x00
<b>SET Rights</b>	0x04

Examples	
HTTP/JSON Get-Request	["get", "name", "nbns_name"]
HTTP/JSON Get-Response	["get", "name", "nbns_name", "0", "pan9320"]
HTTP/JSON Set-Request	["set", "name", "nbns_name", "pan9320"]
HTTP/JSON Set-Response	["set", "name", "nbns_name", "0", "pan9320"]
CMD-UART Get-Request	get name nbns_name\x0d\x0a
CMD-UART Get-Response	get name nbns_name 0 pan9320\x0d\x0a
CMD-UART Set-Request	set name nbns_name pan9320\x0d\x0a
CMD-UART Set-Response	set name nbns_name 0 pan9320\x0d\x0a

### 3.2.6 NetBIOS Mode (Variable: nbns\_mode)

Definition	
<b>Command Option</b>	Get/set
<b>Parameter 1</b>	<p><b>&lt;mode&gt;</b> Enables/disables NBNS mode.</p> <p><b>Default value</b> on</p> <p><b>Range</b> [on, off]</p> <p><b>on:</b> mode enabled</p> <p><b>off:</b> mode disabled</p>
<b>Description</b>	The NBNS mode is used to switch the NBNS service on or off.
<b>GET Rights</b>	0x00
<b>SET Rights</b>	0x04

Examples	
HTTP/JSON Get-Request	<code>["get", "name", "nbns_mode"]</code>
HTTP/JSON Get-Response	<code>["get", "name", "nbns_mode", "0", "on"]</code>
HTTP/JSON Set-Request	<code>["set", "name", "nbns_mode", "on"]</code>
HTTP/JSON Set-Response	<code>["set", "name", "nbns_mode", "0", "on"]</code>
CMD-UART Get-Request	<code>get name nbns_mode\x0d\x0a</code>
CMD-UART Get-Response	<code>get name nbns_mode 0 on\x0d\x0a</code>
CMD-UART Set-Request	<code>set name nbns_mode on\x0d\x0a</code>
CMD-UART Set-Response	<code>set name nbns_mode 0 on\x0d\x0a</code>



### 3.3 NET Service (Module: net)

#### 3.3.1 IP Configuration (Variable: ipcfg)

Definition	
<b>Command Option</b>	get
<b>Parameter 1</b>	<p><b>&lt;interface&gt;</b> Network interface</p> <p><b>Default value</b> -</p> <p><b>Range</b> [0, 1]</p>
<b>Parameter 2</b>	<p><b>&lt;local_ip&gt;</b> IP address</p> <p><b>Default value</b> -</p> <p><b>Range</b> -</p>
<b>Parameter 3</b>	<p><b>&lt;subnet_ip&gt;</b> Subnet mask</p> <p><b>Default value</b> -</p> <p><b>Range</b> -</p>
<b>Parameter 4</b>	<p><b>&lt;gateway_ip&gt;</b> Gateway address</p> <p><b>Default value</b> -</p> <p><b>Range</b> -</p>
<b>Parameter 5</b>	<p><b>&lt;dns1_ip&gt;</b> DNS-Server address</p> <p><b>Default value</b> -</p> <p><b>Range</b> -</p>
<b>Description</b>	<p>IP configuration of an interface (0, 1).</p> <p>Only the first parameter (interface) has to be indicated for a Get request. The Get response in turn includes all five parameters.</p>
<b>GET Rights</b>	0x00

Examples	
HTTP/JSON Get-Request	<code>["get", "net", "ipcfg", "0"]</code>
HTTP/JSON Get-Response	<code>["get", "net", "ipcfg", "0", "0", "192.168.130.30", "255.255.255.0", "192.168.130.1", "192.168.130.1"]</code>
CMD-UART Get-Request	<code>get net ipcfg 0\x0d\x0a</code>
CMD-UART Get-Response	<code>get net ipcfg 0 0 192.168.130.30 255.255.255.0 192.168.130.1 192.168.130.1\x0d\x0a</code>

### 3.3.2 Fixed IP Address (Variable: fixedip)

Definition	
<b>Command Option</b>	Get/set
<b>Parameter 1</b>	<p><b>&lt;interface&gt;</b> Network interface</p> <p><b>Default value</b> 1</p> <p><b>Range</b> [0, 1]</p> <p>0: currently only setting the 'fixedip' of interface 1 supported 1: -</p>
<b>Parameter 2</b>	<p><b>&lt;local_ip&gt;</b> IP address</p> <p><b>Default value</b> -</p> <p><b>Range</b> -</p>
<b>Parameter 3</b>	<p><b>&lt;subnet_ip&gt;</b> Subnet mask</p> <p><b>Default value</b> -</p> <p><b>Range</b> -</p>
<b>Parameter 4</b>	<p><b>&lt;gateway_ip&gt;</b> Gateway address</p> <p><b>Default value</b> -</p> <p><b>Range</b> -</p>
<b>Parameter 5</b>	<p><b>&lt;dns1_ip&gt;</b> DNS server address</p> <p><b>Default value</b> -</p> <p><b>Range</b> -</p>
<b>Description</b>	<p>IP configuration of an interface(0, 1) for the fixed IP mode (0 = Access Point, 1 = Station).</p> <p>Only the first parameter (interface) has to be indicated for a Get request. The Get response in turn includes all five parameters.</p>
<b>GET Rights</b>	0x00
<b>SET Rights</b>	0x04

Examples	
HTTP/JSON Get-Request	["get", "net", "fixedip", "1"]
HTTP/JSON Get-Response	["get", "net", "fixedip", "0", "1", "192.168.130.30", "255.255.255.0", "192.168.130.1", "192.168.130.1"]
HTTP/JSON Set-Request	["set", "net", "fixedip", "1", "192.168.130.30", "255.255.255.0", "192.168.130.1", "192.168.130.1"]
HTTP/JSON Set-Response	["set", "net", "fixedip", "0", "1", "192.168.130.30", "255.255.255.0", "192.168.130.1", "192.168.130.1"]
CMD-UART Get-Request	get net fixedip 1\x0d\x0a
CMD-UART Get-Response	get net fixedip 0 1 192.168.130.30 255.255.255.0 192.168.130.1 192.168.130.1\x0d\x0a
CMD-UART Set-Request	set net fixedip 1 192.168.130.30 255.255.255.0 192.168.130.1 192.168.130.1\x0d\x0a
CMD-UART Set-Response	set net fixedip 0 1 192.168.130.30 255.255.255.0 192.168.130.1 192.168.130.1\x0d\x0a

### 3.3.3 IP Mode (Variable: mode)

Definition	
<b>Command Option</b>	Get/set
<b>Parameter 1</b>	<p><b>&lt;interface&gt;</b> Network interface</p> <p><b>Default value</b> 1</p> <p><b>Range</b> [0, 1] 0: Currently not supported 1: -</p>
<b>Parameter 2</b>	<p><b>&lt;mode&gt;</b> IP Mode</p> <p><b>Default value</b> off without any connection aip+dhcp for the first connection</p> <p><b>Range</b> [aip+dhcp, fip] aip+dhcp: AutoIP and DHCP fip: Fixed-IP</p>
<b>Description</b>	IP mode for the network interface.
<b>GET Rights</b>	0x00
<b>SET Rights</b>	0x04

Examples	
HTTP/JSON Get-Request	["get", "net", "mode", "1"]
HTTP/JSON Get-Response	["get", "net", "mode", "0", "1", "aip+dhcp"]
HTTP/JSON Set-Request	["set", "net", "mode", "1", "aip+dhcp"]
HTTP/JSON Set-Response	["set", "net", "mode", "0", "1", "aip+dhcp"]
CMD-UART Get-Request	get net mode 1\x0d\x0a
CMD-UART Get-Response	get net mode 0 1 aip+dhcp\x0d\x0a
CMD-UART Set-Request	set net mode 1 aip+dhcp\x0d\x0a
CMD-UART Set-Response	set net mode 0 1 aip+dhcp\x0d\x0a

### 3.3.4 PING Request (Variable: ping\_req)

Definition	
<b>Command Option</b>	Set
<b>Parameter 1</b>	<p><b>&lt;ip-adresse&gt;</b> Destination IP address</p> <p><b>Default value</b> -</p> <p><b>Range</b> -</p>
<b>Description</b>	The ping request is only the command to trigger a ping to an IP address. The status of this request must be requested separately.
<b>SET Rights</b>	0x04

Examples	
HTTP/JSON Set-Request	["set", "net", "ping_req", "192.168.1.1"]
HTTP/JSON Set-Response	["set", "net", "ping_req", "0", "192.168.1.1"]
CMD-UART Set-Request	set net ping_req 192.168.1.1\x0d\x0a
CMD-UART Set-Response	set net ping_req 0 192.168.1.1\x0d\x0a

### 3.3.5 PING Status (Variable: ping\_status)

Definition	
<b>Command Option</b>	Get
<b>Parameter 1</b>	<p><b>&lt;ip-address&gt;</b> Destination IP address</p> <p><b>Default value</b> -</p> <p><b>Range</b> -</p>
<b>Parameter 2</b>	<p><b>&lt;status&gt;</b> Status of the ping request</p> <p><b>Default value</b> -</p> <p><b>Range</b> [0-]</p> <p>0: Idle</p> <p>1: Ping sending</p> <p>2: Ping success</p> <p>3: Ping error</p>
<b>Parameter 3</b>	<p><b>&lt;time&gt;</b> Response time for the ping request in [ms]</p> <p><b>Default value</b> -</p> <p><b>Range</b> -</p>
<b>Description</b>	Status of a ping request.
<b>GET Rights</b>	0x01

Examples	
HTTP/JSON Get-Request	<code>["get","net","ping_status"]</code>
HTTP/JSON Get-Response	<code>["get","net","ping_status","0","192.168.1.1","1","65"]</code>
CMD-UART Get-Request	<code>get net ping_status\x0d\x0a</code>
CMD-UART Get-Response	<code>get net ping_status 0 192.168.1.1 1 65\x0d\x0a</code>

## 3.4 Email (Module: smail)

### 3.4.1 User Mail Address (Variable: sender)

Definition	
<b>Command Option</b>	Get/set
<b>Parameter 1</b>	<p><b>&lt;address&gt;</b> Email address of the sender.</p> <p><b>Default value</b> -</p> <p><b>Range</b> String with a maximum of 60 characters</p>
<b>Description</b>	Mail address of the sender.
<b>GET Rights</b>	0x01
<b>SET Rights</b>	0x04

Examples	
HTTP/JSON Get-Request	["get", "smaill", "sender"]
HTTP/JSON Get-Response	["get", "smaill", "sender", "0", "maxmuster@muster.de"]
HTTP/JSON Set-Request	["set", "smaill", "sender", "maxmuster@muster.de"]
HTTP/JSON Set-Response	["set", "smaill", "sender", "0", "maxmuster@muster.de"]
CMD-UART Get-Request	get smaill sender\x0d\x0a
CMD-UART Get-Response	get smaill sender 0 maxmuster@muster.de\x0d\x0a
CMD-UART Set-Request	set smaill sender maxmuster@muster.de\x0d\x0a
CMD-UART Set-Response	set smaill sender 0 maxmuster@muster.de\x0d\x0a

### 3.4.2 Server Configuration (Variable: server)

Definition	
<b>Command Option</b>	Get/set
<b>Parameter 1</b>	<p><b>&lt;server_name&gt;</b> Name of the email server.</p> <p><b>Default value</b> -</p> <p><b>Range</b> String with a maximum of 31 characters and a minimum of 4 characters.</p>
<b>Parameter 2</b>	<p><b>&lt;port&gt;</b> Port of the server.</p> <p><b>Default value</b> 25</p> <p><b>Range</b> 0-65535</p>
<b>Parameter 3</b>	<p><b>&lt;login&gt;</b> Login for the server.</p> <p><b>Default value</b> -</p> <p><b>Range</b> String with a maximum of 31 characters.</p>
<b>Parameter 4</b>	<p><b>&lt;password&gt;</b> Password for the login.</p> <p><b>Default value</b> -</p> <p><b>Range</b> String with a maximum of 15 characters.</p>
<b>Description</b>	Server configuration for email.
<b>GET Rights</b>	0x01
<b>SET Rights</b>	0x04

Examples	
HTTP/JSON Get-Request	<code>["get","smaill","server"]</code>
HTTP/JSON Get-Response	<code>["get","smaill","server","0","MyServ","25","MyLogin"]</code>
HTTP/JSON Set-Request	<code>["set","smaill","server","MyServ","25","MyLogin","MyPassword"]</code>
HTTP/JSON Set-Response	<code>["set","smaill","server","0","MyServ","25","MyLogin","MyPassword"]</code>
CMD-UART Get-Request	<code>get smaill server\x0d\x0a</code>
CMD-UART Get-Response	<code>get smaill server 0 MyServ 25 MyLogin\x0d\x0a</code>
CMD-UART Set-Request	<code>set smaill server MyServ 25 MyLogin MyPassword\x0d\x0a</code>
CMD-UART Set-Response	<code>set smaill server 0 MyServ 25 MyLogin MyPassword\x0d\x0a</code>

### 3.4.3 Mail Sending (Variable: send)

Definition	
<b>Command Option</b>	set
<b>Parameter 1</b>	<p><b>&lt;recipient&gt;</b> Recipient for the email</p> <p><b>Default value</b> -</p> <p><b>Range</b> String with a maximum of 48 characters.</p>
<b>Parameter 2</b>	<p><b>&lt;subject&gt;</b> Subject of the email.</p> <p><b>Default value</b> -</p> <p><b>Range</b> String with a maximum of 60 characters.</p>
<b>Parameter 3</b>	<p><b>&lt;text&gt;</b> Text of the email.</p> <p><b>Default value</b> -</p> <p><b>Range</b> String with a maximum of 250 characters.</p>
<b>Description</b>	Server configuration for email. The mail text only supports printable ASCII characters. A carriage return new line command is not supported.
<b>SET Rights</b>	0x04

Examples	
HTTP/JSON Set-Request	<code>["set","smaill","send","receiver@web.de","subject","mail"]</code>
HTTP/JSON Set-Response	<code>["set","smaill","send","0","receiver@web.de","subject","mail"]</code>
CMD-UART Set-Request	<code>set smaill send receiver@web.de subject mail\x0d\x0a</code>
CMD-UART Set-Response	<code>set smaill send 0 receiver@web.de subject mail\x0d\x0a</code>

### 3.4.4 Mail Status (Variable: status)

Definition	
<b>Command Option</b>	Get
<b>Parameter 1</b>	<p><b>&lt;status&gt;</b>            Status of the mail module</p> <p><b>Default value</b>        0</p> <p><b>Range</b>                 ⇨ Mail Module Status</p>
<b>Description</b>	Status of the mail module.
<b>GET Rights</b>	0x01

Examples	
HTTP/JSON Get-Request	["get","smail","status"]
HTTP/JSON Get-Response	["get","smail","status","0","1"]
CMD-UART Get-Request	get smail status\x0d\x0a
CMD-UART Get-Response	get smail status 0 1\x0d\x0a

### 3.4.5 Mail Error (Variable: error)

Definition	
<b>Command Option</b>	Get
<b>Parameter 1</b>	<p><b>&lt;value&gt;</b>            Value of the error</p> <p><b>Default value</b>        0</p> <p><b>Range</b>                [0-1]</p> <p>                          0: No error active</p> <p>                          1: Error active</p>
<b>Description</b>	Error value of the email module.
<b>GET Rights</b>	0x01

Examples	
HTTP/JSON Get-Request	["get","smail","error"]
HTTP/JSON Get-Response	["get","smail","error","0","0"]
CMD-UART Get-Request	get smail error\x0d\x0a
CMD-UART Get-Response	get smail error 0 0\x0d\x0a



### 3.5 System (Module: system)

#### 3.5.1 Firmware Version (Variable: firmware)

Definition	
Command Option	Get
Description	Returns the firmware version.
GET Rights	0x00

Examples	
HTTP/JSON Get-Request	<code>["get","system","firmware"]</code>
HTTP/JSON Get-Response	<code>["get","system","firmware","0","V_01_01_02"]</code>
CMD-UART Get-Request	<code>get system firmware\x0d\x0a</code>
CMD-UART Get-Response	<code>get system firmware 0 V_01_01_02\x0d\x0a</code>

#### 3.5.2 MAC Address (Variable: macaddr)

Definition	
Command Option	Get
Description	Returns the MAC address.
GET Rights	0x00

Examples	
HTTP/JSON Get-Request	<code>["get","system","macaddr"]</code>
HTTP/JSON Get-Response	<code>["get","system","macaddr","0","0f:0e:03:02:01:00"]</code>
CMD-UART Get-Request	<code>get system macaddr\x0d\x0a</code>
CMD-UART Get-Response	<code>get system macaddr 0 0f:0e:03:02:01:00\x0d\x0a</code>

#### 3.5.3 Serial Number (Variable: serialnum)

Definition	
Command Option	Get
Description	Returns the serial number.
GET Rights	0x00

Examples	
HTTP/JSON Get-Request	["get","system","serialnum"]
HTTP/JSON Get-Response	["get","system","serialnum","0","Ser00001"]
CMD-UART Get-Request	get system serialnum\x0d\x0a
CMD-UART Get-Response	get system serialnum 0 Ser00001\x0d\x0a

### 3.5.4 Wi-Fi Firmware Version (Variable: wifi\_ver)

Definition	
<b>Command Option</b>	Get
<b>Description</b>	Returns the firmware version of the Wi-Fi module.
<b>GET Rights</b>	0x00

Examples	
HTTP/JSON Get-Request	["get","system","wifi_ver"]
HTTP/JSON Get-Response	["get","system","wifi_ver","0","V_03_02_01"]
CMD-UART Get-Request	get system wifi_ver\x0d\x0a
CMD-UART Get-Response	get system wifi_ver 0 V_03_02_01\x0d\x0a

### 3.5.5 Bootloader Version (Variable: bootl\_ver)

Definition	
<b>Command Option</b>	Get
<b>Description</b>	Returns the bootloader version of the Wi-Fi module.
<b>GET Rights</b>	0x00

Examples	
HTTP/JSON Get-Request	["get","system","bootl_ver"]
HTTP/JSON Get-Response	["get","system","bootl_ver","0","V_01_02_01"]
CMD-UART Get-Request	get system bootl_ver\x0d\x0a
CMD-UART Get-Response	get system bootl_ver 0 V_01_02_01\x0d\x0a

### 3.5.6 Hardware Revision (Variable: hwrev)

Definition	
<b>Command Option</b>	Get
<b>Description</b>	Returns the hardware revision number.
<b>GET Rights</b>	0x00

Examples	
HTTP/JSON Get-Request	["get","system","hwrev"]
HTTP/JSON Get-Response	["get","system","hwrev","0","Rev.01"]
CMD-UART Get-Request	get system hwrev\x0d\x0a
CMD-UART Get-Response	get system hwrev 0 Rev.01\x0d\x0a

### 3.5.7 Restart (Variable: restart)

Definition	
<b>Command Option</b>	Set
<b>Parameter</b>	< - > - <b>Range</b> -
<b>Description</b>	This parameter triggers a restart of the hardware.
<b>SET Rights</b>	0x04

Examples	
HTTP/JSON Set-Request	["set","system","restart"]
HTTP/JSON Set-Response	["set","system","restart","0"]
CMD-UART Set-Request	set system restart\x0d\x0a
CMD-UART Set-Response	set system restart 0\x0d\x0a

### 3.5.8 Reset (Variable: factory)

Definition	
<b>Command Option</b>	Set
<b>Parameter</b>	< - > - <b>Range</b> -
<b>Description</b>	This parameter performs a factory reset of the software. The hardware is restarted after the factory reset.
<b>SET Rights</b>	0x04

Examples	
HTTP/JSON Set-Request	["set","system","factory"]
HTTP/JSON Set-Response	["set","system","factory","0"]
CMD-UART Set-Request	set system factory\x0d\x0a
CMD-UART Set-Response	set system factory 0\x0d\x0a

### 3.5.9 Save Mode (Variable: savemode)

Definition	
<b>Command Option</b>	Set
<b>Parameter 1</b>	<p><b>&lt;savemode&gt;</b>      The save mode can be set to „manual“ or „automatic“.</p> <p><b>Default value</b>      auto</p> <p><b>Range</b>                auto or manual</p>
<b>Description</b>	<p>„Manual“ means that the user has to trigger a save of the current configuration.</p> <p>„Automatic“ means that the configuration is automatically saved after every change.</p>
<b>SET Rights</b>	0x04

Examples	
HTTP/JSON Set-Request	<code>["set", "system", "savemode", "auto"]</code>
HTTP/JSON Set-Response	<code>["set", "system", "savemode", "0", "auto"]</code>
HTTP/JSON Get-Request	<code>["get", "system", "savemode"]</code>
HTTP/JSON Get-Response	<code>["get", "system", "savemode", "0", "auto"]</code>
CMD-UART Set-Request	<code>set system savemode auto\x0d\x0a</code>
CMD-UART Set-Response	<code>set system savemode 0 auto\x0d\x0a</code>
CMD-UART Get-Request	<code>get system savemode\x0d\x0a</code>
CMD-UART Get-Response	<code>get system savemode 0 auto\x0d\x0a</code>

### 3.5.10 Save the Configuration (Variable: savecfg)

Definition	
<b>Command Option</b>	Set
<b>Parameter 1</b>	<p><b>&lt; - &gt;</b>                -</p> <p><b>Range</b>                -</p>
<b>Description</b>	This command triggers saving of the configuration.
<b>GET Rights</b>	0x02

Examples	
HTTP/JSON Set-Request	<code>["set", "system", "savecfg"]</code>
HTTP/JSON Set-Response	<code>["set", "system", "savecfg", "0"]</code>
CMD-UART Set-Request	<code>set system savecfg\x0d\x0a</code>
CMD-UART Set-Response	<code>set system savecfg 0\x0d\x0a</code>

## 3.5.11 Power Save Mode (Variable: psm)

Definition	
<b>Command Option</b>	Set
<b>Parameter 1</b>	<p><b>&lt;mode&gt;</b> Power save mode</p> <p><b>Default value</b> -</p> <p><b>Range</b> [0,1,2] 0 = Disable power save mode 1 = IEEE power save modus 2 = Power down</p>
<b>Parameter 2</b>	<p><b>&lt;dtims&gt;</b> Multiple dtims (16 bit)</p> <p><b>Default value</b> -</p> <p><b>Range</b> [0...65535] 65534: CLOSEST_DTIM_TO_LISTEN_INTERVAL)</p>
<b>Parameter 3</b>	<p><b>&lt;BeaconTimeout&gt;</b> Beacon miss timeout (16 bit)</p> <p><b>Default value</b> -</p> <p><b>Range</b> [0...65535]</p>
<b>Parameter 4 (only for mode 1)</b>	<p><b>&lt;ListInt&gt;</b> Local listen interval (16 bit)</p> <p><b>Default value</b> -</p> <p><b>Range</b> [0...65535]</p>
<b>Parameter 5 (only for mode 1)</b>	<p><b>&lt;AdhocWakePeriod&gt;</b> Adhoc wake period (16 bit)</p> <p><b>Default value</b> -</p> <p><b>Range</b> [0...65535]</p>
<b>Parameter 6 (only for mode 1)</b>	<p><b>&lt;delay&gt;</b> Delay to powersave (16 bit)</p> <p><b>Default value</b> -</p> <p><b>Range</b> [0...65535]</p>
<b>Description</b>	Sets the MC200's and the radio chip's power save mode.
<b>SET Rights</b>	0x04

Examples	
HTTP/JSON Set-Request	<code>["set", "system", "psm", "1"]</code>
HTTP/JSON Set-Response	<code>["set", "system", "psm", "0", "1"]</code>
CMD-UART Set-Request	<code>set system psm 1 \x0d\x0a</code>
CMD-UART Set-Response	<code>set system psm 0 \x0d\x0a</code>
CMD-UART Set-Request	<code>set system psm 1 65534 5 0 0 0 \x0d\x0a</code>
CMD-UART Set-Response	<code>set system psm 0 1 65534 5 0 0 0 \x0d\x0a</code>

### 3.6 User Management (Module: user)

#### 3.6.1 User Edit/Add (Variable: edit)

Definition	
<b>Command Option</b>	Set
<b>Parameter 1</b>	<p><b>&lt;index#&gt;</b> User index number</p> <p><b>Default value</b> -</p> <p><b>Range</b> [0-4]</p>
<b>Parameter 2</b>	<p><b>&lt;name&gt;</b> Login name</p> <p><b>Default value</b> -</p> <p><b>Range</b> String with a maximum of 29 characters</p>
<b>Parameter 3</b>	<p><b>&lt;password&gt;</b> Password for login</p> <p><b>Default value</b> -</p> <p><b>Range</b> String with a maximum of 29 characters</p>
<b>Parameter 4</b>	<p><b>&lt;rights&gt;</b> Rights of the user</p> <p><b>Default value</b> -</p> <p><b>Range</b> [0..255] ⇒ <a href="#">User Rights</a></p>
<b>Description</b>	Sets the user with login name, password, and rights.
<b>SET Rights</b>	0x80

Examples	
HTTP/JSON Set-Request	<code>["set","user","edit","1","Max","password","61"]</code>
HTTP/JSON Set-Response	<code>["set","user","edit","0","1","Max","password","61"]</code>
CMD-UART Set-Request	<code>set user edit 1 Max password 61\x0d\x0a</code>
CMD-UART Set-Response	<code>set user edit 0 1 Max password 61\x0d\x0a</code>

### 3.6.2 User Name (Variable: name)

Definition	
<b>Command Option</b>	Get
<b>Parameter 1</b>	<p><b>&lt;index#&gt;</b> User index number</p> <p><b>Default value</b> -</p> <p><b>Range</b> [0..4]</p>
<b>Parameter 2</b>	<p><b>&lt;name#&gt;</b> User name</p> <p><b>Default value</b> -</p> <p><b>Range</b> -</p>
<b>Parameter 3</b>	<p><b>&lt;rights#&gt;</b> User rights</p> <p><b>Default value</b> -</p> <p><b>Range</b> [0..255] ⇒ <a href="#">User Rights</a></p>
<b>Description</b>	Returns the user name for the login.
<b>GET Rights</b>	0x01

Examples	
HTTP/JSON Get-Request	["get","user","name","1"]
HTTP/JSON Get-Response	["get","user","name","0","1","Max","61"]
CMD-UART Get-Request	get user name 1\x0d\x0a
CMD-UART Get-Response	get user name 0 1 Max 61\x0d\x0a

### 3.6.3 User Delete (Variable: delete)

Definition	
<b>Command Option</b>	Set
<b>Parameter 1</b>	<p><b>&lt;index#&gt;</b> User index number</p> <p><b>Default value</b> -</p> <p><b>Range</b> [1-4]</p>
<b>Description</b>	Deletes a user. The user 0 cannot be deleted.
<b>SET Rights</b>	0x80

Examples	
HTTP/JSON Set-Request	["set","user","delete","1"]
HTTP/JSON Set-Response	["set","user","delete","0","1"]
CMD-UART Set-Request	set user delete 1\x0d\x0a
CMD-UART Set-Response	set user delete 0 1\x0d\x0a

### 3.6.4 Default Login Active (Variable: dlogin)

Definition	
<b>Command Option</b>	get
<b>Description</b>	Returns '0', if the default login is still set. Returns '1', if the login has been changed.
<b>GET Rights</b>	0x01

Examples	
HTTP/JSON Get-Request	["get", "user", "dlogin"]
HTTP/JSON Get-Response	["get", "user", "dlogin", "0", "1"]
CMD-UART Get-Request	get user dlogin\x0d\x0a
CMD-UART Get-Response	get user dlogin 0 1\x0d\x0a

### 3.6.5 Default Rights (Variable: drights)

Definition	
<b>Command Option</b>	Get/set
<b>Parameter 1</b>	<p><b>&lt;drights&gt;</b>                      Parameter default rights if no authentication was requested</p> <p><b>Default value</b>                    0</p> <p><b>Range</b>                                0 - 255</p>
<b>Description</b>	This parameter is used for default rights, if no authentication is needed for the files.
<b>GET Rights</b>	0x01
<b>SET Rights</b>	0x80

Examples	
HTTP/JSON Get-Request	["get", "user", "drights"]
HTTP/JSON Get-Response	["get", "user", "drights", "0", "0"]
CMD-UART Get-Request	get user drights\x0d\x0a
CMD-UART Get-Response	get user drights 0 0\x0d\x0a
HTTP/JSON Set-Request	["set", "user", "drights", "1"]
HTTP/JSON Set-Response	["set", "user", "drights", "0", "1"]
CMD-UART Set-Request	set user drights 1\x0d\x0a
CMD-UART Set-Response	set user drights 0 1\x0d\x0a



### 3.6.6 My Rights (Variable: myrights)

Definition	
<b>Command Option</b>	Get
<b>Parameter 1</b>	<p><b>&lt;myrights&gt;</b> Returns the current rights</p> <p><b>Default value</b> 0</p> <p><b>Range</b> 0 - 255</p>
<b>Description</b>	This parameter is used to display the current rights for a logged in user.
<b>GET Rights</b>	0x00

Examples	
HTTP/JSON Get-Request	<code>["get", "user", "myrights"]</code>
HTTP/JSON Get-Response	<code>["get", "user", "myrights", "0", "3"]</code>
CMD-UART Get-Request	<code>get user myrights\x0d\x0a</code>
CMD-UART Get-Response	<code>get user myrights 0 3\x0d\x0a</code>

## 3.7 Telnet (Module: telnet)

### 3.7.1 Mode (Variable: mode)

Definition	
<b>Command Option</b>	Get/set
<b>Parameter 1</b>	<p><b>&lt;mode&gt;</b> Mode for telnet</p> <p><b>Default value</b> on</p> <p><b>Range</b> 'on' or 'off'</p>
<b>Description</b>	Returns the Telnet mode. This parameter can also set the Telnet mode to 'on' or 'off'.
<b>GET Rights</b>	0x01
<b>SET Rights</b>	0x04

Examples	
HTTP/JSON Get-Request	<code>["get", "telnet", "mode"]</code>
HTTP/JSON Get-Response	<code>["get", "telnet", "mode", "0", "on"]</code>
CMD-UART Get-Request	<code>get telnet mode\x0d\x0a</code>
CMD-UART Get-Response	<code>get telnet mode 0 on\x0d\x0a</code>
HTTP/JSON Set-Request	<code>["set", "telnet", "mode", "off"]</code>
HTTP/JSON Set-Response	<code>["set", "telnet", "mode", "0", "off"]</code>
CMD-UART Set-Request	<code>set telnet mode off\x0d\x0a</code>
CMD-UART Set-Response	<code>set telnet mode 0 off\x0d\x0a</code>

### 3.7.2 Auth (Variable: auth)

Definition	
<b>Command Option</b>	Get/set
<b>Parameter 1</b>	<p><b>&lt;auth&gt;</b> Authentication for Telnet enabled/disabled</p> <p><b>Default value</b> on</p> <p><b>Range</b> 'on' or 'off'</p>
<b>Description</b>	The Telnet authentication can be enabled or disabled with this parameter.
<b>GET Rights</b>	0x01
<b>SET Rights</b>	0x80

Examples	
HTTP/JSON Get-Request	["get", "telnet", "auth"]
HTTP/JSON Get-Response	["get", "telnet", "auth", "0", "on"]
CMD-UART Get-Request	get telnet auth\x0d\x0a
CMD-UART Get-Response	get telnet auth 0 on\x0d\x0a
HTTP/JSON Set-Request	["set", "telnet", "auth", "off"]
HTTP/JSON Set-Response	["set", "telnet", "auth", "0", "off"]
CMD-UART Set-Request	set telnet auth off\x0d\x0a
CMD-UART Set-Response	set telnet auth 0 off\x0d\x0a

## 3.8 Firmware Update (Module: fwu)

### 3.8.1 Progress (Variable: progress)

Definition	
<b>Command Option</b>	Get
<b>Parameter 1</b>	<p><b>&lt;progress&gt;</b> The firmware update progress.</p> <p><b>Default value</b> -</p> <p><b>Range</b> [0..100] in %</p>
<b>Description</b>	This command returns the progress of the firmware update process.
<b>GET Rights</b>	0x01

Examples	
HTTP/JSON Get-Request	["get", "fwu", "progress"]
HTTP/JSON Get-Response	["get", "fwu", "progress", "0", "50"]
CMD-UART Get-Request	get fwu progress\x0d\x0a
CMD-UART Get-Response	get fwu progress 0 50\x0d\x0a

### 3.8.2 Status (Variable: status)

Definition	
<b>Command Option</b>	Get
<b>Parameter 1</b>	<p><b>&lt;status&gt;</b> The firmware update status.</p> <p><b>Default value</b> 0</p> <p><b>Range</b> ⇒ <a href="#">Firmware Update Status List</a></p>
<b>Description</b>	Returns the status of the firmware update process.
<b>GET Rights</b>	0x01

Examples	
HTTP/JSON Get-Request	<code>["get", "fwu", "status"]</code>
HTTP/JSON Get-Response	<code>["get", "fwu", "status", "0", "0"]</code>
CMD-UART Get-Request	<code>get fwu status\x0d\x0a</code>
CMD-UART Get-Response	<code>get fwu status 0 0\x0d\x0a</code>

### 3.8.3 Check for Firmware Version (Variable: checkfw)

Definition	
<b>Command Option</b>	Get/set
<b>Parameter 1</b>	<p><b>&lt;firmware&gt;</b> Only returned in a Get request; answer from the firmware update server</p> <p><b>Default value</b> -</p> <p><b>Range</b> -</p>
<b>Description</b>	This command returns an information from the firmware update server. The firmware update server is responsible for the answer. With the Set request, a request is triggered to get the current firmware version on the server. With the Get request, the answer from the server can be requested.
<b>GET Rights</b>	0x01
<b>SET Rights</b>	0x04

Examples	
HTTP/JSON Get-Request	<code>["get", "fwu", "checkfw"]</code>
HTTP/JSON Get-Response	<code>["get", "fwu", "checkfw", "0", "V_01_00_07"]</code>
HTTP/JSON Set-Request	<code>["set", "fwu", "checkfw"]</code>
HTTP/JSON Set-Response	<code>["set", "fwu", "checkfw", "0"]</code>
CMD-UART Get-Request	<code>get fwu checkfw\x0d\x0a</code>
CMD-UART Get-Response	<code>get fwu checkfw 0 V_01_00_07\x0d\x0a</code>
CMD-UART Set-Request	<code>set fwu checkfw\x0d\x0a</code>
CMD-UART Set-Response	<code>set fwu checkfw 0\x0d\x0a</code>

### 3.8.4 Firmware Update (Variable: fwupdate)

Definition	
<b>Command Option</b>	Set
<b>Parameter 1</b>	<p><b>&lt;version&gt;</b>            Version number for the update. If this parameter is empty, the latest version will be installed.</p> <p><b>Default value</b>        -</p> <p><b>Range</b>                 -</p>
<b>Description</b>	A firmware update via HTTP server is executed with this parameter.
<b>SET Rights</b>	0x40

Examples	
HTTP/JSON Set-Request	<code>["set", "fwu", "fwupdate"]</code>
HTTP/JSON Set-Response	<code>["set", "fwu", "fwupdate", "0"]</code>
CMD-UART Set-Request	<code>set fwu fwupdate\x0d\x0a</code>
CMD-UART Set-Response	<code>set fwu fwupdate 0\x0d\x0a</code>

## 3.9 Command UART (Module: cmduart)

### 3.9.1 Send (Variable: send)

Definition	
<b>Command Option</b>	Set
<b>Parameter 1</b>	<p><b>&lt;data&gt;</b>                Data which is sent over the UART to set specific data</p> <p><b>Default value</b>        -</p> <p><b>Range</b></p>
<b>Description</b>	The data is sent over the UART.
<b>SET Rights</b>	0x08 or 0x10

Examples	
HTTP/JSON Get-Request	<code>["set", "cmduart", "send", "example"]</code>
HTTP/JSON Get-Response	<code>["set", "cmduart", "send", "0", "example"]</code>
CMD-UART Get-Request	<code>set cmduart send example\x0d\x0a</code>
CMD-UART Get-Response	<code>set cmduart send 0 example\x0d\x0a</code>

### 3.9.2 Request (Variable: request)

Definition	
<b>Command Option</b>	Get
<b>Parameter 1</b>	<p><b>&lt;data&gt;</b> Data which is sent over the UART to request specific data</p> <p><b>Default value</b> -</p> <p><b>Range</b> -</p>
<b>Description</b>	The data is sent over the UART.
<b>GET Rights</b>	0x08 or 0x10

Examples	
HTTP/JSON Get-Request	<code>["get", "cmduart", "request", "temp"]</code>
HTTP/JSON Get-Response	<code>["get", "cmduart", "request", "0", "temp=100"]</code>
CMD-UART Get-Request	<code>get cmduart request temp\x0d\x0a</code>
CMD-UART Get-Response	<code>get cmduart request 0 temp=100\x0d\x0a</code>

### 3.9.3 UART Configuration (Variable: cfg)

Definition	
<b>Command Option</b>	Set/get
<b>Parameter 1</b>	<p><b>&lt;baudrate&gt;</b> Baudrate for the UART</p> <p><b>Default value</b> 115 200</p> <p><b>Range</b> ⇒ Supported Baud Rates</p>
<b>Parameter 2</b>	<p><b>&lt;data_bits&gt;</b> Data bit for the UART. This parameter cannot be set, it will always be 8.</p> <p><b>Default value</b> 8</p> <p><b>Range</b> [5-8] ⇒ Data Bit Type</p>
<b>Parameter 3</b>	<p><b>&lt;parity&gt;</b> Parity bit for the UART. This parameter cannot be set, it will always be 0.</p> <p><b>Default value</b> 0</p> <p><b>Range</b> [0-4] ⇒ Parity Types</p>
<b>Parameter 4</b>	<p><b>&lt;stopbit&gt;</b> Stop bits for the UART. This parameter cannot be set, it will always be 1.</p> <p><b>Default value</b> 1</p> <p><b>Range</b> [1-2] ⇒ Stop Bit Type</p>
<b>Parameter 5</b>	<p><b>&lt;hwhandshake&gt;</b> Hardware handshake enabled or disabled. The hardware handshake will always be disabled (0) for the command UART.</p> <p><b>Default value</b> 0</p>
<b>Description</b>	UART configuration for the Netcat module.
<b>GET Rights</b>	0x01
<b>SET Rights</b>	0x04

Examples	
HTTP/JSON Get-Request	<code>["get", "cmduart", "cfg"]</code>
HTTP/JSON Get-Response	<code>["get", "cmduart", "cfg", "0", "115200", "8", "0", "1", "0"]</code>
HTTP/JSON Set-Request	<code>["set", "cmduart", "cfg", "115200", "8", "0", "1", "0"]</code>
HTTP/JSON Set-Response	<code>["set", "cmduart", "cfg", "0", "115200", "8", "0", "1", "0"]</code>
CMD-UART Get-Request	<code>get cmduart cfg\x0d\x0a</code>
CMD-UART Get-Response	<code>get cmduart cfg 0 115200 8 0 1 0\x0d\x0a</code>
CMD-UART Set-Request	<code>set cmduart cfg 115200 8 0 1 0\x0d\x0a</code>
CMD-UART Set-Response	<code>set cmduart cfg 0 115200 8 0 1 0\x0d\x0a</code>

### 3.9.4 Send User Rights via UART (Variable: auth\_state)

Definition	
<b>Command Option</b>	Set/get
<b>Parameter 1</b>	<p><b>&lt;auth_state&gt;</b>      Flag for sending additional current user rights via UART</p> <p><b>Default value</b>      off</p> <p><b>Range</b>                on/off</p>
<b>Description</b>	If this flag is set, specific user rights are applied ⇒ <a href="#">Additional User Rights</a> .
<b>GET Rights</b>	0x01
<b>SET Rights</b>	0x80

Examples	
HTTP/JSON Get-Request	<code>["get", "cmduart", "auth_state"]</code>
HTTP/JSON Get-Response	<code>["get", "cmduart", "auth_state", "0", "off"]</code>
HTTP/JSON Set-Request	<code>["set", "cmduart", "auth_state", "off"]</code>
HTTP/JSON Set-Response	<code>["set", "cmduart", "auth_state", "0", "off"]</code>
CMD-UART Get-Request	<code>get cmduart auth_state \x0d\x0a</code>
CMD-UART Get-Response	<code>get cmduart auth_state 0 off\x0d\x0a</code>
CMD-UART Set-Request	<code>set cmduart auth_state off\x0d\x0a</code>
CMD-UART Set-Response	<code>set cmduart auth_state 0 off\x0d\x0a</code>

## 3.10 GPIO (Module: gpio)

### 3.10.1 Mode (Variable: mode)

Definition	
<b>Command Option</b>	Get/set
<b>Parameter 1</b>	<p><b>&lt;pin#&gt;</b>              A pin of the GPIO.</p> <p><b>Default value</b>      -</p> <p><b>Range</b>                ⇒ <a href="#">5.8 GPIO</a></p>
<b>Parameter 2</b>	<p><b>&lt;mode&gt;</b>                Mode of a GPIO pin.</p> <p><b>Default value</b>      out</p> <p><b>Range</b>                [in, out]</p> <p><b>in:</b> input</p> <p><b>out:</b> output</p>
<b>Description</b>	Gets/sets the mode of a specified GPIO pin.
<b>GET Rights</b>	0x01
<b>SET Rights</b>	0x04

Examples	
HTTP/JSON Get-Request	<code>["get","gpio","mode","4"]</code>
HTTP/JSON Get-Response	<code>["get","gpio","mode","0","4","in"]</code>
HTTP/JSON Set-Request	<code>["set","gpio","mode","4","in"]</code>
HTTP/JSON Set-Response	<code>["set","gpio","mode","0","4","in"]</code>
CMD-UART Get-Request	<code>get gpio mode 4\x0d\x0a</code>
CMD-UART Get-Response	<code>get gpio mode 0 4 in\x0d\x0a</code>
CMD-UART Set-Request	<code>set gpio mode 4 in\x0d\x0a</code>
CMD-UART Set-Response	<code>set gpio mode 0 4 in\x0d\x0a</code>

### 3.10.2 State (Variable: state)

Definition	
<b>Command Option</b>	Get/set
<b>Parameter 1</b>	<p><b>&lt;pin#&gt;</b> A pin of the GPIO</p> <p><b>Default value</b> -</p> <p><b>Range</b> ⇒ 5.8 GPIO</p>
<b>Parameter 2</b>	<p><b>&lt;state&gt;</b> Enables/disables a GPIO pin</p> <p><b>Default value</b> off</p> <p><b>Range</b> [on, off]</p> <p><b>on:</b> Enabled</p> <p><b>off:</b> Disabled</p>
<b>Description</b>	Enables/disables a specified GPIO pin.
<b>GET Rights</b>	0x01
<b>SET Rights</b>	0x04

Examples	
HTTP/JSON Get-Request	<code>["get","gpio","state","4"]</code>
HTTP/JSON Get-Response	<code>["get","gpio","state","0","4","off"]</code>
HTTP/JSON Set-Request	<code>["set","gpio","state","4","off"]</code>
HTTP/JSON Set-Response	<code>["set","gpio","state","0","4","off"]</code>
CMD-UART Get-Request	<code>get gpio state 4\x0d\x0a</code>
CMD-UART Get-Response	<code>get gpio state 0 4 off\x0d\x0a</code>
CMD-UART Set-Request	<code>set gpio state 4 off\x0d\x0a</code>
CMD-UART Set-Response	<code>set gpio state 0 4 off\x0d\x0a</code>



### 3.11 HTTP Client (Module: httpc)

#### 3.11.1 Status (Variable: status)

Definition	
<b>Command Option</b>	Get
<b>Parameter 1</b>	<p><b>&lt;status&gt;</b> HTTP client status</p> <p><b>Default value</b> 0</p> <p><b>Range</b> [0-2] ⇒ <a href="#">HTTP Client Status</a></p>
<b>Description</b>	Returns the status of the HTTP client.
<b>GET Rights</b>	0x01

Examples	
HTTP/JSON Get-Request	["get", "httpc", "status"]
HTTP/JSON Get-Response	["get", "httpc", "status", "0", "1"]
CMD-UART Get-Request	get httpc status\x0d\x0a
CMD-UART Get-Response	get httpc status 0 1\x0d\x0a

#### 3.11.2 HTTP Server (Variable: server)

Definition	
<b>Command Option</b>	Get/set
<b>Parameter 1</b>	<p><b>&lt;server-adr&gt;</b> HTTP server address</p> <p><b>Default value</b> -</p> <p><b>Range</b> String with a maximum of 39 characters</p>
<b>Parameter 2</b>	<p><b>&lt;port&gt;</b> HTTP server port number</p> <p><b>Default value</b> -</p> <p><b>Range</b> [0-65535]</p>
<b>Parameter 3</b>	<p><b>&lt;ressource&gt;</b> HTTP server ressource</p> <p><b>Default value</b> -</p> <p><b>Range</b> String with a maximum of 99 characters</p>
<b>Parameter 4</b>	<p><b>&lt;username&gt;</b> HTTP server username</p> <p><b>Default value</b> -</p> <p><b>Range</b> String with a maximum of 29 characters</p>
<b>Parameter 5</b>	<p><b>&lt;password&gt;</b> HTTP server password</p> <p><b>Default value</b> -</p> <p><b>Range</b> String with a maximum of 19 characters</p>
<b>Description</b>	Returns the status of the HTTP client.
<b>GET Rights</b>	0x01
<b>SET Rights</b>	0x04

Examples	
HTTP/JSON Get-Request	["get", "httpc", "server"]
HTTP/JSON Get-Response	["get", "httpc", "server", "0", "ServerAdr", "80", "ressource", "username"]
HTTP/JSON Set-Request	["set", "httpc", "server", "ServerAdr", "80", "ressource", "username", "password"]
HTTP/JSON Set-Response	["set", "httpc", "server", "0", "ServerAdr", "80", "ressource", "username", "password"]
CMD-UART Get-Request	get httpc server\x0d\x0a
CMD-UART Get-Response	get httpc server 0 ServerAdr 80 ressource username\x0d\x0a
CMD-UART Set-Request	set httpc server ServerAdr 80 ressource username password\x0d\x0a
CMD-UART Set-Response	set httpc server 0 ServerAdr 80 ressource username password\x0d\x0a

### 3.11.3 Post (Variable: post)

Definition	
<b>Command Option</b>	Set
<b>Parameter 1</b>	<p><b>&lt;data&gt;</b> Data which should be sent</p> <p><b>Default value</b> -</p> <p><b>Range</b> String with a maximum of 299 characters</p>
<b>Description</b>	Data which should be sent to the server. To send the data, a valid server must be configured and a connection to this server must be available.
<b>GET Rights</b>	0x01
<b>SET Rights</b>	0x04

Examples	
HTTP/JSON Set-Request	["set", "httpc", "post", "data"]
HTTP/JSON Set-Response	["set", "httpc", "post", "0", "data"]
CMD-UART Set-Request	set httpc post data\x0d\x0a
CMD-UART Set-Response	set httpc post 0 data\x0d\x0a

### 3.11.4 Get (Variable: get)

Definition	
<b>Command Option</b>	Get/set
<b>Parameter 1</b>	<p><b>&lt;data&gt;</b> Data which should be sent</p> <p><b>Default value</b> -</p> <p><b>Range</b> String with a maximum of 299 characters</p>
<b>Description</b>	Data which should be sent to the server.
<b>GET Rights</b>	0x01
<b>SET Rights</b>	0x04

Examples	
HTTP/JSON Get-Request	["get", "httpc", "get"]
HTTP/JSON Get-Response	["get", "httpc", "get", "0", "data"]
HTTP/JSON Set-Request	["set", "httpc", "get"]
HTTP/JSON Set-Response	["set", "httpc", "get", "0"]
CMD-UART Get-Request	get httpc get\x0d\x0a
CMD-UART Get-Response	get httpc get 0\x0d\x0a
CMD-UART Set-Request	set httpc get\x0d\x0a
CMD-UART Set-Response	set httpc get 0\x0d\x0a

### 3.11.5 CMDAPI (Variable: cmdapi)

Definition	
<b>Command Option</b>	Set
<b>GET Rights</b>	0x01
<b>SET Rights</b>	0x04

Examples	
HTTP/JSON Get-Request	["set", "httpc", "cmdapi"]
HTTP/JSON Get-Response	["set", "httpc", "cmdapi", "0"]
CMD-UART Get-Request	set httpc cmdapi\x0d\x0a
CMD-UART Get-Response	set httpc cmdapi 0\x0d\x0a

### 3.11.6 Token (Variable: token)

Definition	
<b>Command Option</b>	Get/set
<b>Parameter 1</b>	<p><b>&lt;token&gt;</b> Server token for the client</p> <p><b>Default value</b> -</p> <p><b>Range</b> 19 characters</p>
<b>Description</b>	Token from the server for the client.
<b>GET Rights</b>	0x01
<b>SET Rights</b>	0x04

Examples	
HTTP/JSON Get-Request	["get", "httpc", "token"]
HTTP/JSON Get-Response	["get", "httpc", "token", "0", "12345abc"]
CMD-UART Get-Request	get httpc token\x0d\x0a
CMD-UART Get-Response	get httpc token 0 12345abc\x0d\x0a
HTTP/JSON Set-Request	["set", "httpc", "token", "12345abc"]
HTTP/JSON Set-Response	["set", "httpc", "token", "0", "12345abc"]
CMD-UART Set-Request	set httpc token 12345abc\x0d\x0a
CMD-UART Set-Response	set httpc token 0 12345abc\x0d\x0a

## 3.12 Netcat (Module: netcat)

### 3.12.1 State (Variable: state)

Definition	
<b>Command Option</b>	Get/set
<b>Parameter 1</b>	<p><b>&lt;state&gt;</b> Starts or stops a TCP Listen or a TCP Connect</p> <p><b>Default value</b> off</p> <p><b>Range</b> [on, off]</p>
<b>Description</b>	The variable can start or stop a TCP Listen or a TCP Connect, according to the mode parameter.
<b>GET Rights</b>	0x01
<b>SET Rights</b>	0x04

Examples	
HTTP/JSON Get-Request	["get", "netcat", "state"]
HTTP/JSON Get-Response	["get", "netcat", "state", "0", "on"]
HTTP/JSON Set-Request	["set", "netcat", "state", "off"]
HTTP/JSON Set-Response	["set", "netcat", "post", "0", "off"]
CMD-UART Get-Request	get netcat state\x0d\x0a
CMD-UART Get-Response	get netcat state 0 on\x0d\x0a
CMD-UART Set-Request	set netcat state off\x0d\x0a
CMD-UART Set-Response	set netcat state 0 off\x0d\x0a

### 3.12.2 Server IP (Variable: serv\_ip)

Definition	
<b>Command Option</b>	Get/set
<b>Parameter 1</b>	<p><b>&lt;server-adr&gt;</b> Server IP address</p> <p><b>Default value</b> -</p> <p><b>Range</b> -</p>
<b>Description</b>	Server IP address for a TCP Connect with the Netcat module.
<b>GET Rights</b>	0x01
<b>SET Rights</b>	0x04

Examples	
HTTP/JSON Get-Request	["get", "netcat", "serv_ip"]
HTTP/JSON Get-Response	["get", "netcat", "serv_ip", "0", "125.69.12.1"]
HTTP/JSON Set-Request	["set", "netcat", "serv_ip", "125.69.12.1"]
HTTP/JSON Set-Response	["set", "netcat", "serv_ip", "0", "125.69.12.1"]
CMD-UART Get-Request	get netcat serv_ip\x0d\x0a
CMD-UART Get-Response	get netcat serv_ip 0 125.69.12.1\x0d\x0a
CMD-UART Set-Request	set netcat serv_ip 125.69.12.1\x0d\x0a
CMD-UART Set-Response	set netcat serv_ip 0 125.69.12.1\x0d\x0a

### 3.12.3 Server Login (Variable: serv\_login)

Definition	
<b>Command Option</b>	Set/get
<b>Parameter 1</b>	<p><b>&lt;username&gt;</b> Username on the server for the login</p> <p><b>Default value</b> -</p> <p><b>Range</b> String with a maximum of 29 characters</p>
<b>Parameter 2</b>	<p><b>&lt;password&gt;</b> Password on the server for the login. The password will not be returned with a get request.</p> <p><b>Default value</b> -</p> <p><b>Range</b> String with a maximum of 29 characters</p>
<b>Description</b>	Login for the server for Netcat.
<b>GET Rights</b>	0x01
<b>SET Rights</b>	0x04

Examples	
HTTP/JSON Get-Request	<code>["get", "netcat", "serv_login"]</code>
HTTP/JSON Get-Response	<code>["get", "netcat", "serv_login", "0", "username"]</code>
HTTP/JSON Set-Request	<code>["set", "netcat", "serv_login", "username", "password"]</code>
HTTP/JSON Set-Response	<code>["set", "netcat", "serv_login", "0", "username", "password"]</code>
CMD-UART Get-Request	<code>get netcat serv_login\x0d\x0a</code>
CMD-UART Get-Response	<code>get netcat serv_login 0 username\x0d\x0a</code>
CMD-UART Set-Request	<code>set netcat serv_login username password\x0d\x0a</code>
CMD-UART Set-Response	<code>set netcat serv_login 0 username password\x0d\x0a</code>

### 3.12.4 Telnet Option (Variable: telopt)

Definition	
<b>Command Option</b>	Set/get
<b>Parameter 1</b>	<p><b>&lt;option&gt;</b> on or off</p> <p><b>Default value</b> off</p> <p><b>Range</b> on/off</p>
<b>Description</b>	Enables or disables the Telnet option for Netcat. All Telnet options ⇒ <a href="#">3.7 Telnet (Module: telnet)</a> .
<b>GET Rights</b>	0x01
<b>SET Rights</b>	0x04

Examples	
HTTP/JSON Get-Request	["get","netcat","telopt"]
HTTP/JSON Get-Response	["get","netcat","telopt","0","on"]
HTTP/JSON Set-Request	["set","netcat","telopt","off"]
HTTP/JSON Set-Response	["set","netcat","telopt","0","off"]
CMD-UART Get-Request	get netcat telopt\x0d\x0a
CMD-UART Get-Response	get netcat telopt 0 on\x0d\x0a
CMD-UART Set-Request	set netcat telopt off\x0d\x0a
CMD-UART Set-Response	set netcat telopt 0 off\x0d\x0a

### 3.12.5 TCP Port (Variable: tcp\_port)

Definition	
<b>Command Option</b>	Set/get
<b>Parameter 1</b>	<p><b>&lt;port&gt;</b> Port for a TCP connection (listen and connect)</p> <p><b>Default value</b> -</p> <p><b>Range</b> 1-65535</p>
<b>Description</b>	TCP port for a TCP Listen or a TCP Connect
<b>GET Rights</b>	0x01
<b>SET Rights</b>	0x04

Examples	
HTTP/JSON Get-Request	["get","netcat","tcp_port"]
HTTP/JSON Get-Response	["get","netcat","tcp_port","0","12345"]
HTTP/JSON Set-Request	["set","netcat","tcp_port","12345"]
HTTP/JSON Set-Response	["set","netcat","tcp_port","0","12345"]
CMD-UART Get-Request	get netcat tcp_port\x0d\x0a
CMD-UART Get-Response	get netcat tcp_port 0 12345\x0d\x0a
CMD-UART Set-Request	set netcat tcp_port 12345\x0d\x0a
CMD-UART Set-Response	set netcat tcp_port 0 12345\x0d\x0a

### 3.12.6 Mode (Variable: mode)

Definition	
<b>Command Option</b>	Set/get
<b>Parameter 1</b>	<p><b>&lt;mode&gt;</b>            client or server</p> <p><b>Default value</b>        client</p> <p><b>Range</b>                client or server</p>
<b>Description</b>	The mode can be set to "client" or "server". In the client mode, the netcat module connects to a server. In the server mode, the Netcat module is the server and a client can establish a connection.
<b>GET Rights</b>	0x01
<b>SET Rights</b>	0x04

Examples	
HTTP/JSON Get-Request	["get", "netcat", "mode"]
HTTP/JSON Get-Response	["get", "netcat", "mode", "0", "client"]
HTTP/JSON Set-Request	["set", "netcat", "mode", "server"]
HTTP/JSON Set-Response	["set", "netcat", "mode", "0", "server"]
CMD-UART Get-Request	get netcat mode\x0d\x0a
CMD-UART Get-Response	get netcat mode 0 client\x0d\x0a
CMD-UART Set-Request	set netcat mode server\x0d\x0a
CMD-UART Set-Response	set netcat mode 0 server\x0d\x0a

### 3.12.7 Exclusive Lock (Variable: lock)

Definition	
<b>Command Option</b>	Get
<b>Parameter 1</b>	<p><b>&lt;lock ID&gt;</b>            Handle ID for the exclusive lock</p> <p><b>Default value</b>        -</p> <p><b>Range</b>                0 - 65535</p>
<b>Description</b>	To lock the communication, a user can request a handle ID. This handle ID must be added to the write request or the read request.
<b>GET Rights</b>	0x01

Examples	
HTTP/JSON Get-Request	["get", "netcat", "lock"]
HTTP/JSON Get-Response	["get", "netcat", "lock", "0", "15252"]
CMD-UART Get-Request	get netcat lock\x0d\x0a
CMD-UART Get-Response	get netcat lock 0 15252\x0d\x0a



### 3.12.8 Release Lock (Variable: release)

Definition	
<b>Command Option</b>	Set
<b>Parameter 1</b>	<p><b>&lt;lock ID&gt;</b> Handle ID to remove the exclusive lock</p> <p><b>Default value</b> -</p> <p><b>Range</b> 0 - 65535</p>
<b>Description</b>	Removes the exclusive lock, which was created with the “lock” parameter. The lock will also be released after five minutes without any data activity.
<b>SET Rights</b>	0x04

Examples	
HTTP/JSON Set-Request	<code>["set", "netcat", "release", "15252"]</code>
HTTP/JSON Set-Response	<code>["set", "netcat", "release", "0", "15252"]</code>
CMD-UART Set-Request	<code>set netcat release 15252\x0d\x0a</code>
CMD-UART Set-Response	<code>set netcat release 0 15252\x0d\x0a</code>

### 3.12.9 Authentication (Variable: auth)

Definition	
<b>Command Option</b>	Set/get
<b>Parameter 1</b>	<p><b>&lt;auth&gt;</b> on/off</p> <p><b>Default value</b> on</p> <p><b>Range</b> on or off</p>
<b>Description</b>	Enables or disables the authentication for netcat.
<b>GET Rights</b>	0x01
<b>SET Rights</b>	0x80

Examples	
HTTP/JSON Get-Request	<code>["get", "netcat", "auth"]</code>
HTTP/JSON Get-Response	<code>["get", "netcat", "auth", "0", "on"]</code>
HTTP/JSON Set-Request	<code>["set", "netcat", "auth", "off"]</code>
HTTP/JSON Set-Response	<code>["set", "netcat", "auth", "0", "off"]</code>
CMD-UART Get-Request	<code>get netcat auth\x0d\x0a</code>
CMD-UART Get-Response	<code>get netcat auth 0 on\x0d\x0a</code>
CMD-UART Set-Request	<code>set netcat auth off\x0d\x0a</code>
CMD-UART Set-Response	<code>set netcat auth 0 off\x0d\x0a</code>

### 3.12.10 DTR Pin (Variable: dtrpin)

Definition	
<b>Command Option</b>	Set/get
<b>Parameter 1</b>	<p><b>&lt;pin&gt;</b> DTR pin for the netcat UART</p> <p><b>Default value</b> -</p> <p><b>Range</b> Available pins ⇨ GPIO</p>
<b>Description</b>	<p>Set the DTR pin for the netcat UART. This Pin is set to low or high. The dtr-pin is a pin which can be configured via telnet-option. The dtr pin can be set to high or low via a telnet command.</p>
<b>GET Rights</b>	0x01
<b>SET Rights</b>	0x04

Examples	
HTTP/JSON Get-Request	<code>["get", "netcat", "dtrpin"]</code>
HTTP/JSON Get-Response	<code>["get", "netcat", "dtrpin", "0", "4"]</code>
HTTP/JSON Set-Request	<code>["set", "netcat", "dtrpin", "4"]</code>
HTTP/JSON Set-Response	<code>["set", "netcat", "dtrpin", "0", "4"]</code>
CMD-UART Get-Request	<code>get netcat dtrpin \x0d\x0a</code>
CMD-UART Get-Response	<code>get netcat dtrpin 0 4 \x0d\x0a</code>
CMD-UART Set-Request	<code>set netcat dtrpin 4 \x0d\x0a</code>
CMD-UART Set-Response	<code>set netcat dtrpin 0 4 \x0d\x0a</code>

### 3.13 UART Binary (Module: binuart)

#### 3.13.1 Uart Configuration (Variable: cfg)

Definition	
<b>Command Option</b>	set/get
<b>Parameter 1</b>	<p><b>&lt;baudrate&gt;</b> Baudrate for the UART</p> <p><b>Default value</b> 115 200</p> <p><b>Range</b> ⇒ Supported Baud Rates</p>
<b>Parameter 2</b>	<p><b>&lt;data_bits&gt;</b> Data bit for the UART, this parameter cannot be set, it will always be 8.</p> <p><b>Default value</b> 8</p> <p><b>Range</b> [5-8] ⇒ Data Bit Type</p>
<b>Parameter 3</b>	<p><b>&lt;parity&gt;</b> Parity bit for the UART, this parameter cannot be set, it will always be 0.</p> <p><b>Default value</b> 0</p> <p><b>Range</b> [0-4] ⇒ Parity Types</p>
<b>Parameter 4</b>	<p><b>&lt;stopbit&gt;</b> Stopbits for the UART. This parameter cannot be set, it will always be 1.</p> <p><b>Default value</b> 1</p> <p><b>Range</b> [1-2] ⇒ Stop Bit Type</p>
<b>Parameter 5</b>	<p><b>&lt;hwhandshake&gt;</b> Hardware handshake enabled or disabled</p> <p><b>Default value</b> 1</p> <p><b>Range</b> [0-1]</p>
<b>Description</b>	Uart configuration for the netcat module.
<b>GET Rights</b>	0x01
<b>SET Rights</b>	0x04

Examples	
HTTP/JSON Get-Request	["get", "binuart", "cfg"]
HTTP/JSON Get-Response	["get", "binuart", "cfg", "0", "115200", "8", "0", "1", "1"]
HTTP/JSON Set-Request	["set", "binuart", "cfg", "115200", "8", "0", "1", "1"]
HTTP/JSON Set-Response	["set", "binuart", "cfg", "0", "115200", "8", "0", "1", "1"]
CMD-UART Get-Request	get binuart cfg\x0d\x0a
CMD-UART Get-Response	get binuart cfg 0 115200 8 0 1 1\x0d\x0a
CMD-UART Set-Request	set binuart cfg 115200 8 0 1 1\x0d\x0a
CMD-UART Set-Response	set binuart cfg 0 115200 8 0 1 1\x0d\x0a

### 3.13.2 Read Data (Variable: data)

Definition	
<b>Command Option</b>	read
<b>Parameter 1</b>	<p><b>&lt;data&gt;</b> Data which is received</p> <p><b>Default value</b> -</p> <p><b>Range</b> Maximum of 150 binary values as two ASCII-hex values</p>
<b>Parameter 2</b>	<p><b>&lt;lock ID&gt;</b> Handle ID</p> <p><b>Default value</b> -</p> <p><b>Range</b> 0 - 65535</p>
<b>Description</b>	Reads out the binary UART. If there is no exclusive lock (no handle ID), the parameter can be set to "0" or "".
<b>READ Rights</b>	0x20

Examples	
HTTP/JSON Read-Request	<code>["read", "binuart", "data", "1234"]</code>
HTTP/JSON Read-Response	<code>["read", "binuart", "data", "0", 1234, "data"]</code>
CMD-UART Read-Request	<code>read binuart data 1234\x0d\x0a</code>
CMD-UART Read-Response	<code>read binuart data 0 1234 data\x0d\x0a</code>

### 3.13.3 Write Data (Variable: data)

Definition	
<b>Command Option</b>	Write
<b>Parameter 1</b>	<p><b>&lt;lock ID&gt;</b> Handle ID</p> <p><b>Default value</b> -</p> <p><b>Range</b> 0 - 65535</p>
<b>Parameter 2</b>	<p><b>&lt;data&gt;</b> Data to write</p> <p><b>Default value</b> -</p> <p><b>Range</b> Maximum of 300 characters</p>
<b>Description</b>	Write data to the UART. If there is no exclusive lock (no handle ID), the parameter must be set to "0".
<b>WRITE Rights</b>	0x20

Examples	
HTTP/JSON Write-Request	<code>["write", "binuart", "data", "1234", "64656667"]</code>
HTTP/JSON Write-Response	<code>["write", "binuart", "data", "1234", "0", "64656667"]</code>
CMD-UART Write-Request	<code>write binuart data 1234 64656667\x0d\x0a</code>
CMD-UART Write-Response	<code>write binuart data 0 1234 64656667\x0d\x0a</code>

### 3.14 Time (Module: time)

#### 3.14.1 UTC (Variable: utc)

Definition	
<b>Command Option</b>	Get
<b>Parameter 1</b>	<p><b>&lt;utc&gt;</b> Date and time of the UTC</p> <p><b>Default value</b> -</p> <p><b>Range</b> UTC</p>
<b>Description</b>	<p>UTC with the ISO 8601 time format: Year-Month-DayTHour:Minute:Seconds 2015-08-31T11:12:13</p>
<b>GET Rights</b>	0x00

Examples	
HTTP/JSON Get-Request	<code>["get", "time", "utc"]</code>
HTTP/JSON Get-Response	<code>["get", "time", "utc", "0", "2015-08-31T11:12:13"]</code>
CMD-UART Get-Request	<code>get time utc\x0d\x0a</code>
CMD-UART Get-Response	<code>get time utc 2015-08-31T11:12:13\x0d\x0a</code>

#### 3.14.2 Time Sync (Variable: sync)

Definition	
<b>Command Option</b>	Get/set
<b>Parameter 1</b>	<p><b>&lt;sync&gt;</b> Timeout value for NTP time update</p> <p><b>Default value</b> 3 600 seconds</p> <p><b>Range</b> 0 -&gt; no NTP update</p> <p><b>Range</b> Minimum: 3 600 seconds Maximum: 4 294 967 294 seconds</p>
<b>Description</b>	Sets the timeout for NTP updates. Setting the value to 0 results in no updates for the time being.
<b>GET Rights</b>	0x01

<b>Examples</b>	
HTTP/JSON Get-Request	["get","time","sync"]
HTTP/JSON Get-Response	["get","time","sync","0","3600"]
HTTP/JSON Set-Request	["set","time","sync","3600"]
HTTP/JSON Set-Response	["set","time","sync","0","3600"]
CMD-UART Get-Request	get time utc\x0d\x0a
CMD-UART Get-Response	get time utc 0 3600\x0d\x0a
CMD-UART Set-Request	set time utc 3600\x0d\x0a
CMD-UART Set-Response	set time utc 0 3600\x0d\x0a

## 4 Communication with the Host Controller

The command UART is the UART 0. The command UART knows two commands, the „request“ and the „send“ command.

### 4.1 Command UART

#### Additional User Rights

With the command ⇒ [3.9.4 Send User Rights via UART \(Variable: auth\\_state\)](#), an application can decide to send the current user rights via a command UART command.

The user rights + '#' in front of the rights, will be added into the data string, in front of the data divided by a space from the original data.

Examples	
Original data (binary)	54 65 73 74
Original data (ASCII)	Test
Added user rights to the data (binary)	23 66 66 20 54 65 73 74
Added user rights to the data (ASCII)	#ff Test

### 4.2 Binary UART

#### Supported Baud Rates

Baud								
300	600	1 200	2 400	4 800	9 600	14 400	19 200	28 800
38 400	56 000	57 600	115 200	128 000	256 000	520 000	780 000	1 500 000

## 5 Status Information

### 5.1 Telegram Return Code

The telegram returns a return code with every GET or SET request. The return code can be delivered in two different ways.

- The return code is delivered with an ERROR telegram ⇒ [Return Code Error Telegram](#).
- The return code is delivered in the normal telegram after the variable ⇒ [Return Code Info](#) and [Return Code for Normal Telegram](#).

#### Return Code Info

Return Code	Description
0	Nor error detected
10	Info that the binary UART still has data left
11	Warning: Data overflow on the binary UART

#### Return Code Error Telegram

Return Code	Description
100	Empty telegram
101	Command not found. Only get or set command is valid.
102	Module not found
103	Variable not found
104	Data error
105	Internal buffer error

#### Return Code for Normal Telegram

Return Code	Description
201	No permission to get or set this parameter
202	Get request is not possible for this variable
203	Set request is not possible for this variable
204	Parameter error
205	Format error
206	Variable not found
207	Communication error with the UART
208	Internal Storage error
301	The parameter length is too long for the variable
302	The application is not yet ready
303	Netcat wrong exclusive lock ID



Examples for Error Telegram	
HTTP/JSON Get-Request	["test", "system", "firmware"]
HTTP/JSON Get-Response	["ERROR", "101"]
CMD-UART Get-Request	test system firmware\x0d\x0a
CMD-UART Get-Response	ERROR 101 "Command not found"\x0d\x0a

Examples for a Normal Telegram Return Code	
HTTP/JSON Set-Request	["set", "wlan", "infra_mode", "test"]
HTTP/JSON Set-Response	["set", "wlan", "infra_mode", "204", "test"]
CMD-UART Set-Request	set wlan infra_mode test\x0d\x0a
CMD-UART Set-Response	set wlan infra_mode 204 test\x0d\x0a

## 5.2 Wi-Fi Parameter

### Wi-Fi Status Information

Status Number	Description
0	Not initialized
1	Not connected to a network
2	Establishing a connection
3	IP address is obtained
4	Connected to the network
5	Searching for networks
6	Entered network not found
7	Authentication failed

## 5.3 Mail Service

### Mail Module Status

Status Number	Description
0	Not configured
1	Configuration ok - ready to send
2	Mail is sending
3	Failed to connect to server
4	Authentication failed

## 5.4 User Management

### User Rights

Rights	Description
0x80	Rights to change the user configuration update
0x40	Rights to do a firmware update
0x20	Rights to SET/GET data via BINUART
0x10	Rights to SET/GET data via CMDUART HIGH
0x08	Rights to SET/GET data via CMDUART LOW
0x04	Rights to SET parameters with a high priority
0x02	Rights to SET parameters with a low priority
0x01	Rights to GET parameters

### The PAN9320 has three demo users:

Group	Description
Admin (0xFF)	<ul style="list-style-type: none"> <li>• Rights to change the user configuration update</li> <li>• Rights to do a firmware update</li> <li>• Rights to SET/GET data via BINUART</li> <li>• Rights to SET/GET data via CMDUART HIGH</li> <li>• Rights to SET/GET data via CMDUART LOW</li> <li>• Rights to SET parameters with a high priority</li> <li>• Rights to SET parameters with a low priority</li> <li>• Rights to GET parameters with a low priority</li> </ul>
Technician (0x3F)	<ul style="list-style-type: none"> <li>• Rights to SET/GET data via BINUART</li> <li>• Rights to SET/GET data via CMDUART HIGH</li> <li>• Rights to SET/GET data via CMDUART LOW</li> <li>• Rights to SET parameters with a high priority</li> <li>• Rights to SET parameters with a low priority</li> <li>• Rights to GET parameters with a high priority</li> <li>• Rights to GET parameters with a low priority</li> </ul>
User (0x0B)	<ul style="list-style-type: none"> <li>• Rights to SET/GET data via CMDUART LOW</li> <li>• Rights to SET parameters with a low priority</li> <li>• Rights to GET parameters with a low priority</li> </ul>

## 5.5 Firmware Update

### Firmware Update Status List

Number	Description
0	Ready for update
1	Update active
2	Checking for new firmware
3	Update done
4	Update failed

## 5.6 HTTP Client

### HTTP Client Status

Number	Description
0	Not configured
1	Ready to send data
2	Data is sending
3	Error

## 5.7 UART Configuration

### 5.7.1 Data Bit Type

Number	Description
8	8 bit data mode

### 5.7.2 Parity Types

Number	Description
0	No parity

### 5.7.3 Stop Bit Type

Number	Description
1	One Stop Bits Select

## 5.8 GPIO

The PAN9320 module offers some GPIOs which can be controlled via the command API.

The following table shows all available pins.

**GPIO Pin Table**

GPIO CMD Index	HW Pin Name	HW Pin Description
4	GPIO4	Digital I/O #4
5	GPIO5	Digital I/O #5
6	GPIO6	Digital I/O #6
7	GPIO7	Digital I/O #7
28	GPIO28	Digital I/O #28
44	GPIO44	Digital I/O #44
45	GPIO45	Digital I/O #45

## 6 Contact Details

### 6.1 Contact Us

Please contact your local Panasonic Sales office for details on additional product options and services:

For Panasonic Sales assistance in the **EU**, visit

<https://eu.industrial.panasonic.com/about-us/contact-us>

Email: [wireless@eu.panasonic.com](mailto:wireless@eu.panasonic.com)

For Panasonic Sales assistance in **North America**, visit the Panasonic Sales & Support Tool to find assistance near you at

<https://na.industrial.panasonic.com/distributors>

Please visit the **Panasonic Wireless Technical Forum** to submit a question at

<https://forum.na.industrial.panasonic.com>

### 6.2 Product Information

Please refer to the Panasonic Wireless Connectivity website for further information on our products and related documents:

For complete Panasonic product details in the **EU**, visit

<http://pideu.panasonic.de/products/wireless-modules.html>

For complete Panasonic product details in **North America**, visit

<http://www.panasonic.com/rfmodules>