Version 4.0
Application Note

ads-tec IIT GmbH

# IRF1000 IRF2000 IRF3000

## Application Note - ads-tec allgemeine API Spezifikation und Übersicht

## Document History

| | | |
|---|---|---|
| 1.0 | 12/08/2013 | Initial Version |
| 1.1 | 21/03/2014 | WLAN Kapitel hinzugefügt |
| 1.2 | 25/03/2014 | 1.7.3 Zertifikatsupload Hinweis |
| 1.3 | 29/04/2014 | Update für IRF2000 2.5.1 |
| 2.0 | 27/10/2014 | Aktualisierung der Adresse |
| 2.1 | 21/04/2015 | Neuer Abschnitt 1.7.4, System Reboot, Firmware Update im Hintergrund |
| 2.2 | 20/10/2015 | Erweiterung um meminfo und java_thread_dump |
| 2.3 | 04/02/2016 | Erweiterung um Netzwerkdiagnosefunktionen, CustomerSettings |
| 2.4 | 16/11/2016 | Erweiterung der Big-LinX-Statusabfragen um den wwh_status |
| 2.5 | 23/11/2016 | Erweiterung der 3G/UMTS-Statusabfragen |
| 2.6 | 08/05/2017 | Erweiterung um GNSS/GPS Statusabfragen |
| 2.7 | 31/07/2017 | Erweiterung um Big-LinX autopin |
| 2.8 | 20/09/2017 | Erweiterung der 3G/UMTS-Statusabfragen für LTE |
| 2.9 | 03/07/2018 | Erweiterung um DNS Konfiguration und Big-LinX Details |
| 3.0 | 03/12/2019 | Erweiterung um Mobilfunk-Bändauswahl |
| 3.1 | 18/02/2022 | Erweiterung um Netzwerkdiagnosefunktionen |
| 3.2 | 08/03/2022 | Erweiterung um Definition der Big-LinX Zustandsnamen |
| 4.0 | 12/04/2023 | Konsolidierung und Aktualisierung |

Das Original dieser Betriebsanleitung wurde in deutscher Sprache verfasst. Jede nicht deutschsprachige Ausgabe dieser Betriebsanleitung ist eine Übersetzung der deutschen Betriebsanleitung.

# Einleitung

Alle ads-tec Produkte der Serien IRF1000, IRF2000 oder IRF3000 stellen verschiedene APIs bereit um mit den Geräten zu kommunizieren. Alle dieser drei APIs teilen eine gemeinsame Konfigurations-Datenbank Zugriff und dieselbe Status Abfragen und Zustandsinformationen abzurufen.

Die folgende Tabelle listet die verschiedenen APIs anhand der Firmwareversion und Produktserie:

| | classic http API | adsdp API | JSON RPC 2.0 API |
|---|---|---|---|
| **IRF2000** | x | x | ab 2.5.0 |
| **IRF1000** | x | x | x |
| **IRF3000** | x | x | x |

Für Neuentwicklungen wird empfohlen die JSON RPC 2.0 API zu verwenden, da diese die Neuste und Leistungsfähigste in der Auswahl darstellt. Die anderen beiden sind dennoch eine gute Wahl für spezielle Anforderungen:

- classic http API: Ist besonders leicht zu verwenden mit Kommandozeilen http Clients wie „curl“ oder „wget“ und ist perfekt geeignet für einfache Batch-Konfigurationstools. Weiterführende Informationen finden sich him Application Note “ads-tec HTTP API specification”.
- adsdp API: Ist die einzige API die auf Ethernet-Ebene arbeitet zusätzlich zu TCP. Daher ist es hiermit möglich ads-tec Geräte im Netzwerk zu detektieren und ihre IP-Adressen abzufragen und IP-Adresskonflikte aufzulösen. Diese API ist als einfaches TLV-Protokoll designt, so das eine Implementierung in SPSen oder eigenen Programmierumgebungen ohne High-Level Schnittstellen wie http oder JSON. Weiterführende Informationen finden sich im Application Note “ads-tec adsp API specification“.
- JSON RPC 2.0 API: Es existieren hier zusätzliche Objekte und Methoden, die über die hier beschriebenen Dinge wie Status und Konfiguration hinausgehen. Weiterführende Informationen finden sich im Application Note “ads-tec JSON RPC 2.0 API specification.

Hinweis: Die in diesem Dokument aufgeführten Variablen und Aufrufe sind in keinem Fall vollständig, die Geräte verfügen intern über wesentlich mehr davon. Ads-tec versucht jedoch diese Auswahl über die Firmwareversionen hinweg stabil zu halten, so dass keine Anpassungen an den jeweiligen Softwarepaketen nötig werden, welche diese APIs verwenden. Falls für Ihre Softwareentwicklung Aufrufe oder Variablen vermissen wenden Sie sich einfach an Ihren jeweiligen Ansprechpartner bei ads-tec.

# Netzwerkschnittstellen Bezeichnungen und andere Konventionen

Die Angaben in den Tabellen dieses Dokumentes sind teilweise in regulären Ausdrücken dargestellt. So steht beispielsweise „(wan|lan|lan_port(1|2|3|4))_proto" für die möglichen Werte "wan_proto",
„lan_proto","lan_port1_proto","lan_port2_proto","lan_port3_proto" und „lan_port4_proto".
An anderer Stelle werden Werte symbolisch beschrieben, z.B. „IP Adresse" – gemeint ist ein eine Ipv4-Adresse in der Punkt-Notation, z.B. „192.168.0.254".
Schnittstellen-Namen tauchen als Parameter in zwei unterschiedlichen Formaten auf:

1. interner Name, findet Verwendung in Variablennamen der Konfiguration
2. Systemname, Name der Schnittstelle auf Betriebssystem-Ebene

Zu beachten ist, dass je nach Operationsmodus verschiedene Schnittstellen vorhanden sind. Die OpenVPN-L2-Schnittstellen sind keine eigenständigen Layer3-Schnittstellen. Sie sind in der Netzwerkbrücke „br0" enthalten, d.h. es zählt die IP-Adresse von „LAN".

# Konfiguration und Statusabfragen

in diesem Kapitel werden einige wichtige Variablen der internen Konfigurationsdatenbank und deren zulässige Werte vorgestellt. Genauere Beschreibungen zu den Einstellungen und Hinweise dazu finden Sie im Handbuch der Industrial Firewall, das Sie auch auf der ads-tec Homepage (http://www.ads-tec.de) herunterladen können.

## 1.1 Allgemeine Netzwerkeinstellungen

### 1.1.1 Config Variablen:

| Name | Wert | Beschreibung |
|---|---|---|
| opmode | (transbridge\|iprouter \|iprouter5port) | Operationsmodus<br><br>*transbridge*: default Modus. Die Firewall operiert auf Layer-2, wie ein „managed Switch“.<br><br>*Iprouter*: Layer-3 Operation zwischen LAN und WAN Ports<br><br>*Iprouter5port*: wie iprouter, aber eigenständige IP-Konfiguration (Layer-3 Operation) für alle Ports.<br><br>Achtung: Die Änderung des Opmode hat einige Seiteneffekte auf andere Konfigurationsoptionen, die danach z.B. nicht mehr oder dann erst gelten. Betroffen sind u.A. Filterregeln und statische Routen. Weitere Informationen finden sich im Handbuch. |
| (wan\|lan\|lan_port(1\|2\|3\|4))_proto | (static\|dhcp\|dhcpstatic\| dhcpovpn\|pppoedyn) | *static*: statische Konfiguration mit den Werten der Variablen „_proto“, „_ipaddr“ und „_netmask“<br><br>*dhcp*: dynamische Konfiguration über DHCP<br><br>*dhcpstatic*: dynamisch per DHCP mit statischem Fallback<br><br>*dhcpovpn*: dynamische IP über einen OpenVPN Tunnel<br><br>*pppoedyn*: PPPOE mit dynamischer IP, für DSL Uplink.<br><br>Default: static |
| (wan\|lan\|lan_port(1\|2\|3\|4))_ipaddr | IP Adresse | z.B. „192.168.0.254“ (statische Konfiguration) |
| (wan\|lan\|lan_port(1\|2\|3\|4))_netmask | Netzmaske | z.B. „255.255.255.0“ (statische Konfiguration) |
| lan_gateway | IP Adresse | default Gateway. Aus historischen Gründen mit „lan“ bezeichnet, gilt aber für alle Schnittstellen. |
| wan_pppoe_userid | Text | Userid entsprechend DSL Zugangsdaten. Aus historischen Gründen mit „wan“ bezeichnet, gilt aber für das jeweilig mit _proto „pppoedyn“ konfigurierte Interface. |
| wan_pppoe_password | Text | Passwort entsprechend DSL Zugangsdaten. Aus historischen Gründen mit „wan“ bezeichnet, gilt aber für das jeweilig mit _proto „pppoedyn“ konfigurierte Interface. |
| splitbridge_port[1-4] | (enabled\|disabled) | Legt für einen Lan-out Port im extended Router Modus fest, ob er mit Lan-in zusammen eine Netzwerkbrücke (Bridge) bilden soll. Default: disabled |
| dns1 | IP Adresse | Primärer DNS Server |
| dns2 | IP Adresse | Sekundärer DNS Server |
| dns3 | IP Adresse | Tertiärer DNS Server |
| dns_domain | Text | Subdomain des lokalen Netzes, z.B. „intranet.company.de“. Wird bei DNS Anfragen automatisch angehängt wenn ein Hostname ohne Domain angefragt wird. |
| dns_request_all | (enabled\|disabled) | Falls aktiv werden alle hinterlegten DNS Server gleichzeitig angefragt. Ansonsten wird mit dem ersten Server gestartet und nur bei Timeouts wird der nächste Server versucht. |
| dns_proxy | (enabled\|disabled) | Aktiviert den DNS Relay Dienst auf dem Gerät |

### 1.1.2 Status Abfragen

| Name | Parameter | Wert | Beschreibung |
|---|---|---|---|
| if_status | (eth1\|eth0\|ixp1\|ixp0.10(1\|2\|3\|4)) | „no link“ oder Details zum Link Status | Physikalischer Status der Schnittstelle. Rückgabe: (no link \| ((negotiated \| no autonegotiation,) (10\|100)baseT(4\|x)?(-(HD\|FD))? (flow-control, )? Link ok) \| invalid interface \| missing parameter) |
| if_mac | (eth1\|eth0\|ixp0\|ixp1\|ixp0.10(1\|2\|3\|4)) | Mac-Adresse | Hardware-Adresse, z.B. „00:CC:90:00:71:01“ |
| if_ip | (eth1\|eth0\|br0\|br1\|ixp0\|ixp1\|ixp0.10(1\|2\|3\|4)) | IP-Adresse | z.B. „192.168.0.254“ |
| If_rxb | (eth1\|eth0\|br0\|br1\|ixp0\|ixp1\|ixp0.10(1\|2\|3\|4)) | | Gesendete Bytes über das Interface |
| If_txb | (eth1\|eth0\|br0\|br1\|ixp0\|ixp1\|ixp0.10(1\|2\|3\|4)) | | Empfangene Bytes über das Interface |
| routes | - | Routing table | Show the routing table of the system |
| nameserver | - | Aktuelle DNS Server | Leerzeichen speparierte Liste aller aktuell verweneter DNS Server |
| ping4 | <Gegenstelle> [<Anzahl Pakete>] | Ping-Statistik | Sendet ICMP Echo Requests an die Gegenstelle und liefert das Resultat zurück |
| traceroute4 | <Gegenstelle> | | Führt Traceroute aus |
| nslookup4 | <Name> | | Führt die DNS-Anfrage aus |
| arping | <Gegenstelle> | (up\|down) | Sendet arp Anfragen an einen Host und gibt an ob dieser up oder down ist |
| tcpping | <Gegenstelle><Port> | (Host is up\|Host is down) | Sendet ein tcp-syn paket an einen Port auf einem Host und prüft ob ein syn-ack zurückkommt |
| dhcp_gateway | - | IP-Adresse | Liefert den aktuellen DHCP Default Gateway |

## 1.2 Datum und Uhrzeit, NTP

### 1.2.1 Datum & Uhrzeit, Config-Variablen

| Name | Wert | Beschreibung |
|---|---|---|
| ntp_server | IP oder Hostname | NTP Server 1 |
| ntp_server2 | IP oder Hostname | NTP Server 2 |
| ntp_server3 | IP oder Hostname | NTP Server 3 |
| ntp_service | (enabled\|disabled) | NTP Client aktivieren |
| ntp_relay | (enabled\|disabled) | NTP Relay aktivieren |
| timezone | Text | Zeitzone z.B. Europe/Berlin |
| year | Jahr | Datum manuell setzen, z.B.: 2012 |
| month | Monat | Datum manuell setzen, z.B.: 01 für Januar |
| day | Tag | Datum manuell setzen, Tag des Monats |
| hour | Stunde | Uhrzeit manuell setzen |
| minute | Minute | Uhrzeit manuell setzen |
| second | Sekunde | Uhrzeit manuell setzen |

### 1.2.2 Datum & Uhrzeit, Status Abfragen

| Name | Parameter | Wert | Beschreibung |
|---|---|---|---|
| date | - | Datum und Uhrzeit | Aktuelle Systemzeit und Zeitzone, z.B. „Fri Oct 26 00:00:00 CEST 2012“ |

# 1.3 OpenVPN-Konfiguration

## 1.3.1 Allgemeine OpenVPN Verbindungs-Konfiguration, Config-Variablen

Hinweis: Die VPN Verbindung 10 entspricht der Big-LinX VPN-Verbindung, dort wird jedoch nur das Feld 10 verwendet!

| Name | Wert | Beschreibung |
|---|---|---|
| vpn_list_*n*<br><br>(*n*: 0-10) | Slash-separierte Verbindungs-Definition | Liste der – maximal 10 definierbaren – unterschiedlichen OpenVPN Verbindungen. In aufsteigender Reihenfolge zu verwenden (Zuerst vpn_list_0).<br>z.B.:<br><br>“OpenVPN/Client/TCP/192.168.10.1:1194/1194/client1.pem/tap0/switched/”<br><br>Feld 1: aktuell immer OpenVPN<br>Feld 2: Client oder Master<br>Feld 3: TCP oder UDP<br>Feld 4: ZielIP:Port:Proto für Client Verbindungen, hier wird erneut UDP/TCP als Proto geführt<br>Feld 5: TCP/UDP Port für Master Verbindungen<br>Feld 6: Name des Zertifikats<br>Feld 7: interner Interface Name, tap*n* wobei n von 0-9 den Index des Eintrags widerspiegelt<br>Feld 8: Modus der Verbindung: active (Permanent an), deactive (Permanent aus), switched (geschalten über CUT oder zukünftig ebenfalls über vpn_switch_on/off Variablen) |
| vpn_proxy | (enabled\|disabled) | Ob der angegebene HTPP Proxy bei OpenVPN Client Verbindungen verwendet werden soll. Default: disabled |
| vpn_proxyip | IP Adresse / Hostname | IP Adresse oder DNS Hostname des Proxies |
| vpn_proxyport | TCP Port | TCP Port des Proxies |
| vpn_proxyauthmeth | (ntlm\|basic\|none) | Authentifizierungsmethode |
| vpn_proxyuser | Username | Benutzername für Proxy-Authentifizierung |
| vpn_proxypass | Password | Passwort für Proxy-Authentifizierung |

## 1.3.2 Client-spezifische OpenVPN Verbindungs-Konfiguration, Config-Variablen

lan_proto muss auf „dhcpovpn“ gesetzt sein damit die folgenden Einstellungen funktionieren.

| Name | Wert | Beschreibung |
|---|---|---|
| vpn_clientpull_status | (enabled\|disabled) | Zusammen mit „lan_proto“ aktiviert dies DHCP auf der durch „vpn_clientpull_dev“ definierten OpenVPN Client-Schnittstelle. Default: disabled |
| vpn_clientpull_routes | (enabled\|disabled) | Aktiviert das automatische beziehen von Routing-Informationen vom Master. Default: disabled |
| vpn_clientpull_dev | tap[0-9] | Schnittstelle für die obige „clientpull“-Optionen gelten |

## 1.3.3 Master-spezifische OpenVPN Verbindungs-Konfiguration, Config-Variablen

| Name | Wert | Beschreibung |
|---|---|---|
| vpn_ippool_status | (enabled\|disabled) | Aktiviert die IP-Konfiguration der Clients durch den Master. Default: disabled |
| vpn_ippool_startip | IP Adresse | Start-Adresse des IP-Bereiches, der für die Clients zur Verfügung steht |
| vpn_ippool_endip | IP Adresse | End-Adresse des IP-Bereiches, der für die Clients zur Verfügung steht |
| vpn_ippool_masterdev | tap[0-9] | Schnittstelle, für die die „ippool“-Optionen gelten |
| vpn_ippool_pushgw | (enabled\|disabled) | Gibt die IP-Adresse des Masters als default-Gateway an den Client. Default: disabled |
| vpn_ippool_pushsubnet | (enabled\|disabled) | Aktiviert die Weitergabe von statischen Routen an den Client. Default: disabled |

### 1.3.4 Manuelle Aktivierung / Deaktivierung , Config-Variablen

setzt eine konfigurierte VPN-Verbindung im Zustand „Switched“ voraus

| Name | Wert | Beschreibung |
|---|---|---|
| vpn_switch_on | [0-10] | Zu aktivierende VPN Verbindung im Zustand „switched“ durch vpn_switch_now |
| vpn_switch_off | [0-10] | Abzuschaltende VPN Verbindung im Zustand „switched“ durch vpn_switch_now |
| vpn_switch_now | beliebig | Trigger Variable die entsprechende VPN Verbindung im Zustand „switched“ auf- oder abbaut, definiert durch vpn_switch_on oder vpn_switch_off.<br>Löscht die Variablen vpn_switch_on und vpn_switch_off im Anschluss. |

### 1.3.5 Status Abfragen

| Name | Parameter | Wert | Beschreibung |
|---|---|---|---|
| vpnconnstat | tap[0-9] | (up\|down) | z.B. “no autonegotiation, 100baseTx-FD flow-control, link ok” |
| vpnclients | tap[0-9] | Zahl | Anzahl Client-Verbindungen |
| clcerts | - | Verzeichnislisting | Es werden alle Client-Zertifikate des Systems angezeigt |
| cacerts | - | Verzeichnislisting | Es werden alle CA-Zertifikate des Systems angezeigt |
| print_cert | certname | Zertifikat-Details | Details zu dem Zertifikat mit dem Namen „certname“ werden ausgegeben. |
| print_cacert | certname | Zertifikat-Details | Details zu dem CA-Zertifikat mit dem Namen „certname“ werden ausgegeben. |

# 1.4 Big-LinX

## 1.4.1 Config-Variablen

| Name | Value | Description |
|---|---|---|
| vpn_ida_proxy | (enabled\|disabled) | enable or disable OpenVPN HTTP Proxy |
| vpn_ida_proxyip | IP or hostname | proxy IP address or hostname |
| vpn_ida_proxyport | TCP Port | proxy TCP port, i.e. 8080 |
| vpn_ida_proxyauth | (ntlm\|basic\|none) | proxy authenticatiom method |
| vpn_ida_proxyuser | Username | user name if proxy authentication is not none |
| vpn_ida_proxypass | Password | password if proxy authentication is not none |
| sc_autopin | (enabled\|disabled) | enable or disable the autopin feature for smart cards |
| sc_savepin | (enabled\|disabled) | Must be set to “enabled” for cards without autopin |
| sc_pin | Pin | PIN of the Big-LinX smartcard |
| vpn_list_10 | “/” seperated list of servers and modes. | The last field can either be “active” or “switched”. In case of “active” a permanent connection is enabled. In case of the default value “switched” the VPN will only be triggered remote by using the WWH channel or locally by using the VPN key or API.<br><br>All other fields are no longer in use! |

## 1.4.2 Config-Variablen Approval Mode

Der Approval Mode oder VPN-Quittierungsmechanismus erlaubt ein 4-Augen Prinzip für den VPN Verbindungsaufbau. Die entfernte Gegenstelle quittiert den Verbindungswunsch aktiv mit Hilfe eines physikalischen Schlüsselschalters oder via API.

| Name | Value | Description |
|---|---|---|
| vpn_approval_mode | (disabled\|vpnswitch\|api) | Falls der Modus “vpnswitch” gesetzt ist, wird die Big-LinX VPN Verbindung nur gestartet, wenn der VPN Schlüsselschalter aktiv ist. Wenn diese nicht aktiv ist bleibt der vpn_state_name im status blxstat (siehe 1.4.3 ) im Zustand „WAITINGFORAPPROVAL” stehen. Im Falle von „api“ kann die Gegenstelle die Variable „vpn_approve_now“ setzen um den Aufbau der Verbindung zu quittieren. |
| vpn_approve_now | beliebig | Änderung dieses Wertes triggert die o.g. Quittierung der VPN Verbindungsaufbaus. |

## 1.4.3 Status Abfagen

Zusätzlich zu den als OpenVPN Verbindung vpn10 abfragbaren Daten kann man die Datenbasis des Big-LinX Diagnose Webseite abrufen, dies beinhaltet dann auch alle Daten zu Zustand von WWH und Smartcard.

| Name | Parameter | Wert | Beschreibung |
|---|---|---|---|
| blxstat | - | JSON Objekt aller Big-LinX Zustands-Daten | Beispielhafte key value Paare des JSON Objekts bei einer Speicherkarte anstatt einer Smartcard:<br><br>cardstate “VPNSC_CS_I2C”<br>tokenlabel “”<br>openvpnscanstate “VPNSC_OPENVPNSCAN_READY”<br>vpn_state_name “”<br>vpn_oldstate_name””<br>vpn_state_desc””<br>vpn_ip””<br>vpn_permanent 0<br>vpn_server_ip “”<br>pintries -1<br>savepin „<br>vpn_ctrl_state 0<br>wwh_lastbeat 0<br>wwh_service „“enabled”<br>wwh_delay 0<br>wwh_error „“<br><br>Der vpn_state_name kann folgende Zustände annehmen:<br><Leerstring>: VPN-Client läuft momentan nicht<br>“OFF”: VPN nicht verbunden |

| | | | |
|---|---|---|---|
| | | | “INITIAL”: VPN Initialisierung<br>“CONNECTING”: Initialisierung der Verbindung<br>“ASSIGN_IP”: IP-Adresszuweisung der virtuellen VPN-Schnittstelle<br>“ADD_ROUTES”: VPN, füge Routen hinzu<br>“CONNECTED”: Verbunden mit dem VPN-Server<br>“RECONNECTING”: Erneuter Verbindungsversuch mit dem VPN-Server<br>“EXITING”: VPN-Verbindung ist abgebrochen, es wird der nächste Server versucht<br>“WAIT”: Warten auf initiale Antwort vom VPN-Server<br>“AUTH”: Authentifizierung am VPN-Server<br>“GET_CONFIG”: Lade Konfiguration vom Server<br>“RESOLVE”: DNS Auflösung des VPN-Servers<br>“TCP_CONNECT”: Verbinde zum VPN-Server per TCP<br>“WAITINGFORAPPROVAL”: Warte auf Freigabe<br>“UNKNOWN”: Nicht definierter Zustand |
| wwh_status | | ERROR,<br>ONLINE,<br>OFFLINE | Zustand der Big-LinX WWH Verbindung. Hat den Wert ERROR wenn der Dienst nicht läuft, auch wenn er deaktiviert ist, keine Smartcard mit valider Pin vorhanden ist oder anderen Ursachen.<br>OFFLINE bedeutet das keine Verbindung aufgebaut werden konnte.<br>ONLINE bedeutet das der Link stabil ist. |
| pintries | | Counter | Anzahl der verbleibenden PIN Versuche bis die Karte gesperrt wird. |

# 1.5 CUT & ALARM Konfiguration

## 1.5.1 Config-Variablen

| Name | Wert | Beschreibung |
|---|---|---|
| (cut\|alarm)signal_mode | (manual\|auto) | Aktiviert die manuelle oder automatische Bestätigung des CUT- bzw. ALARM-Signals. Default: manual |
| (cut\|alarm)signal_timeout | Sekunden | bei automatischer Bestätigung: nach Ablauf der gegebenen Anzahl Sekunden wird das CUT- bzw. ALARM-Signal aufgehoben |
| (cut\|alarm)signal_ack_now | beliebig | Änderung dieses Wertes verursacht sofortige Aufhebung des Signals |
| cutsignal_cut_now | beliebig | Änderung dieses Wertes verursacht setzen des CUT-Signals |
| alarmsignal_alarm_now | beliebig | Änderung dieses Wertes verursacht setzen des ALARM-Signals |
| cutandalrm_reset | (true\|false) | Aktiviert automatische Quittierung des CUT- oder ALARM-Signals wenn es durch einen Eintrag im Client-Monitoring ausgelöst wurde. (cut\|alarm)signal_mode sollte dann auf „manual“ gesetzt sein. Default: false |
| vpn_ovpn_enable_on_cut | (true\|false) | Aktiviert Auf- und Abbau von OpenVPN-Verbindungen des Typs “Switched” bei Änderung des CUT-Signals. Default: false |
| vpn_ovpn_enb_on_cut_type | (enabled\|disabled) | Wenn vpn_ovpn_enable_on_cut = true<br>*enabled*: baut Verbindung auf wenn CUT-Signal aktiviert wird<br>*disabled*: baut Verbindung auf wenn CUT-Signal deaktiviert wird |

## 1.5.2 Status Abfragen

| Name | Parameter | Wert | Beschreibung |
|---|---|---|---|
| alarm | - | (off\|on) | Gibt an, ob das ALARM-Signal an (on) oder aus (off) ist |
| intcut | - | (off\|on) | Gibt an, ob das interne CUT-Signal an (on) oder aus (off) ist |
| extcut | - | (off\|on) | Gibt an, ob das externe CUT-Signal an (on) oder aus (off) ist |

# 1.6 System

## 1.6.1 Config-Variablen

| Name | Wert | Beschreibung |
|---|---|---|
| save_now | beliebig | Änderung dieses Wertes triggert eine Sicherung der aktuellen Konfiguration in den reboot-festen NVRAM-Speicher. Falls save_tosim auf „enabled“ gesetzt ist, wird die Konfiguration auch auf die Speicherkarte kopiert. |
| save_settings_now | beliebig | Änderung dieses Wertes triggert ein Download der aktuellen Einstellungs-Datei. Dabei wird die gespeicherte Konfiguration heruntergeladen und nicht die aktuell Laufende. |
| nvram_mode | (save\|commit) | save: alle Änderungen werden zwar sofort persistent gespeicht, aber erst nach einem Reboot angewandt.<br>Commit: alle Änderungen werden sofort angewandt, sind aber erst nach setzen des „save_now“ –Triggers persistent gespeichert. |
| system_name | Text | Systemname. Default: Produktbezeichnung |
| system_contact | Text | Kontaktinformation |
| systemname_sndyn | (enabled\|disabled) | Legt fest ob die eindeutige Seriennummer des Gerätes als Systemname verwendet werden soll. Default: enabled |
| hostname | Text | Netzwerk-Name. Default: Produktbezeichnung |
| hostname_sndyn | (enabled\|disabled) | Legt fest ob die eindeutige Seriennummer des Gerätes als Hostname verwendet werden soll. Default: enabled |

## 1.6.2 Firmware Update, Config-Variablen

Das System unterstützt verschiedene Methoden um die Firmware zu aktualisieren.

### 1.6.2.1 Update von einem Server

Das Firmwareupdate kann von einem TFTP, FTP oder http Server angezogen werden. Bei diesem Mechanismus werden Komponenten mit hohem RAM Verbrauch zuvor heruntergefahren.

| Name | Wert | Beschreibung |
|---|---|---|
| update_proto | (http\|ftp\|tftp) | Protokoll, über das die Firmware auf das Gerät geladen werden kann |
| update_server | IP Adresse | IP Adresse des Servers, der die Firmware zur Verfügung stellt |
| update_restoredefaults | (enabled\|disabled) | Aktiviert das Rücksetzen des Gerätes in den Auslieferungszustandes nachdem die neue Firmware geladen wurde. Default: disabled |
| fw_update_now | beliebig | Triggert den Firmware-Update Prozess |

#### 1.6.2.2 Update im Hintergrund

Ab IRF2000 2.6.5 wird ein automatisches Update der zweiten Firmware Partition im Hintergrund unterstützt.
Bei diesem Mechanismus werden Komponenten mit hohem RAM Verbrauch nicht heruntergefahren!
Das Gerät wird dann einmal pro Tag den ads-tec Firmwareserver nach Updates fragen und diese dann auf die zweite Partition installieren. Der Zeitpunkt der Abfrage wird zufällig pro Systemstart zu einer vollen Stunde festgelegt.
Ein Reboot in diese neue Firmware findet NICHT statt, dies muss durch den Benutzer erfolgen oder über die API durch die Reboot Variablen. Die Abfrage welche Firmware Version auf der zweiten Partition bereit steht ist ebenfalls unter dem Kapitel Reboot beschrieben.

| Name | Wert | Beschreibung |
|---|---|---|
| fw_update_url | URL für die „Firmware Description File“ | Standardwert ist: http://www.ads-tec.de/fileadmin/ads-tec/download/software/autoupdate/fldf.txt, der Wert kann für eigene Server geändert werden. |
| fw_update_url_bx | URL für die „Firmware Description File“ | URL für den Fall das das Big-LinX VPN aktiv ist. |
| fw_force_update | enabled oder disabled | Dies ist eine Debug Einstellung. Falls der Wert auf enabled gesetzt wird dann wird die Firmware angezogen auch wenn die aktuelle Version bereits aktuell ist. |
| fw_auto_download | enabled oder disabled | Aktiviert diese Funktion |

### 1.6.3 Zertifikats-Upload (CA,CRL, usw.)

Zertifikatsupload können als http Mime/Multipart POST Formular übertragen werden. Im Abschnitt Beispiele findet sich ein Aufruf mit curl als Demonstration.

| Name | Wert | Beschreibung |
|---|---|---|
| upload_certfile_now | beliebig | Triggert die Upload-Verarbeitung |
| | | |
| update_cert_filename | application/octet-stream *oder* Dateinamen | Die eigentliche Datei als Binary Stream oder der vollständige Dateinamen inkl. Pfad in /usr/local/equinox/data |
| filename_password | Text | Falls es sich um eine Passphrase geschützte p12 oder pem Datei handelt, das Passwort zum dechiffrieren. |

### 1.6.4 Reboot und Reboot Timer, Watchdog

Um einen System Neustart auszulösen können folgende Variablen verwendet werden. Der Mechanismus eignet sich auch als Watchdog für Docker basierte Anwendungen. Wird der Wert von reboot_wait zusammen mit einem erneuten reboot_now immer wieder neu gesetzt bevor der Timer abläuft, so hat man einen einfachen Watchdog welcher das System neu startet sollte die eigene Docker basierte Anwendung nicht mehr funktionieren.
(hier sollte mindestens die Firmwareversion IRF2000 2.6.5 verwendet werden)

| Name | Wert | Beschreibung |
|---|---|---|
| reboot_now | beliebig | Triggert den System Reboot, entweder sofort oder falls reboot_wait einen Wert größer 0 hat, nach der entsprechenden Wartezeit. |
| reboot_wait | Minuten als Ziffer | Wartezeit in Minuten bevor der Reboot (welcher durch reboot_now ausgelöst wird) durchgeführt wird.<br>Um einen bereits laufenden Timer abzubrechen kann reboot_wait auf den Wert 0 gesetzt werden ohne ein erneutes reboot_now zu setzen. |
| firmware_switch_now | beliebig | Aktiviert die Umschaltung in die Alternative Firmware bei einem durch reboot_now ausgelösten Reboot. |

### 1.6.5 Status Abfragen

| Name | Parameter | Wert | Beschreibung |
|---|---|---|---|
| redbootserial | - | Text | Seriennummer des Gerätes |
| realproduct | - | Text | Produktbezeichnung, z.B. “IF1110” |
| versionbuild | - | Text | Firmware Version, z.B. “2.0.6 (Build 55290) |
| uptime | - | Text | U.A. Zeit seit letztem Reboot, z.B. „05:33:48 uo 5:34, load average 0.00, 0.00, 0.00“ |
| simstate | - | Text | … os im: keine Speicherkarte vorhanden<br>synced: aktuelle Konfiguration ist auf Speicherkarte gesichert<br>not synced: aktuelle Konfiguration ist nicht auf Speicherkarte gesichert |
| eventlog | - | mehrzeiliger Test | Ausgabe des Eventlogs, mehrzeilig, daher am Besten mit „statuslong“ abrufen und nicht mit „status“ |
| reboot_timer | - | Sekunden als Ziffern | Wartezeit bis zum Reboot durch einen durch reboot_wait /reboot_now |
| imageversion | - | Text | aktuelle Firmwareversion:<br>z.B. Ads-tec/IRF2xxx/2.6.5/SVN-R13781.B-69636 |
| second_imageversion | - | Text | Firmwareversion auf der zweiten Partition:<br>z.B. Ads-tec/IRF2xxx/2.6.5/SVN-R13781.B-69637 |
| firmwareupdate_prepared | - | yes oder no | “yes” falls die Firmnwareversion auf der zweiten Partition aktualisiert wurde |
| memstat | - | Text | Hauptspeicherauslastung des Systems in % |
| meminfo | Prozessname | Text | Liefert die Hauptspeicherbelegung eines angegebenen Prozesses (oder eine Liste falls mehrere Prozesse mit dem gleichen Namen laufen). Der Prozessname kann der Datei diag.txt aus der Gerätekonfiguration entnommen werden. |
| cpustat | - | Text | Liefert die CPU-Last in % |
| customersettings_size | - | | Liefert die Größe der Customer Settings in kB |
| customersettings_timestamp | - | | Liefert das Änderungsdatum der Customer Settings als Unix Timestamp |

# 1.7 SCEP

## 1.7.1 SCEP Config-Variablen

| Name | Wert | Beschreibung |
|---|---|---|
| scep_service | (enabled\|disabled) | Aktiviert den SCEP-Service |
| scep_url | URL | URL der Server API, z.B. „http://scepserver.ads-tec.de/scep/mscep.dll“ |
| scep_sncn | (enabled\|disabled) | Aktiviert die Verwendung der Geräte-Seriennummer (Eindeutige ID) im Common-name Feld des Zertifikat-Requests. Default: enabled |
| scep_subject | Formated Sting (Format siehe Beschreibung) | Formated String= DNS_FIELD“/“ATTRIBUTES<br>DNS_FIELD=”dns:”NAME<br>ATTRIBUTES=(„C=“COUNTRY”, “)?(“ST=”STATE”, “)?(“L=”LOCALITY”, “)?(“O=”ORGANISATION”, “)?(“OU=”ORGANISATIONAL_UNIT)?<br>NAME, STATE, LOCALITY, ORGANISATION, ORGANISATIONAL_UNIT = Text<br>COUNTRY=(DE\|US\|GB\|...)<br>z.B.: “dns:IF1xxx/C=DE, O=ads-tec”<br>Der SCEP Server wird das Feld dns normalerweise für den Common Name (CN) es ausgetellten Zertifikats verwenden |
| scep_autocrl | (enabled\|disabled) | Aktiviert das automatische Beziehen von CRLs, sofern Informationen über deren Erhalt vom SCEP-Server bezogen werden können. |
| scep_autorenew | Anzahl Tage | Falls gesetzt, wird entsprechend viele Tage vor Ablauf der Gültigkeit des SCEP-Zertifikates versucht, ein neues Zertifikat zu erhalten. |
| scep_keybits | Keylänge | Keylänge=(1024\|2048\|3072\|4096)<br>Wählen sie eine größere Länge des Keys für höhere Sicherheit. Abhängig vom CA-Server kann möglicherweise nicht jede Länge verwendet werden. |
| scep_challenge | Text | Manche SCEP-Server verlangen ein Passwort oder „one-time Challenge“, das im Voraus bekannt sein muss |

## 1.7.2 Status Abfragen

| Name | Parameter | Wert | Beschreibung |
|---|---|---|---|

| | | | |
|---|---|---|---|
| scep | - | [0-99] | 0: SCEP Service ist nicht aktiviert<br>10: SCEP Service startet<br>20: SCEP hat beim SCEP-Server die Server-Zertifikate vom CA-Server und dem SCEP-Server nachgefragt und wartet auf eine Antwort.<br>40: SCEP Service wartet auf angeforderte Client-Zertifikate<br>50: SCEP Service hat bereits erfolgreich ein Client-Zertifikat erhalten<br>60: Error |

# 1.8 3G/4G, UMTS/LTE

Auch wenn die API hier “umts” als Prefix verwendet, alle Calls sind auch für die LTE Variante gültig und kompatibel.

## 1.8.1 3G/4G Config-Variablen

| Name | Wert | Beschreibung | |
|---|---|---|---|
| umts_service | (enabled\|disabled) | Aktiviert den SCEP-Service | |
| umts_apn | Text | APN des Providers, z.B. web.vodafone.de etc. | |
| umts_user | Text | Provider Username, falls benötigt | |
| umts_pass | Text | Provider Passwort, falls benötigt | |
| umts_apn2 | Text | Fallback APN des Providers, z.B. web.vodafone.de etc. (nur über API erreichbar) | Diese drei Variablen können als Fallback-Zugang gesetzt werden. Falls mit den primären Zugangsdaten keine Verbindung zustande kommt wird auf diese hier umgeschaltet.<br><br>(Verfügbar auf IRF2000 ab Version 2.5.1) |
| umts_user2 | Text | Fallback Provider Username, falls benötigt (nur über API erreichbar) | |
| umts_pass2 | Text | Fallback Provider Passwort, falls benötigt (nur über API erreichbar) | |
| umts_dns | (enabled\|disabled) | DNS Server über 3G beziehen. | |
| umts_pin | Text | PIN der SIM Karte | |
| umts_permalink | (enabled\|disabled) | Halte eine permanente Online-Verbindung | |
| umts_puk | Text | PUK der SIM Karte, kann verwendet werden falls die Karte gesperrt wurde. | |
| umts_ondemand | (enabled\|disabled) | Aktiviert das On Demand Dialing, umts_permalink muss dann „disabled“ sein | |
| umts_connect_now | beliebig | Trigger Variable für manuelles Anstoßen der UMTS Verbindung. Änderung des Wertes aktiviert den Vorgang. Nur bei ondemand = enabled | |
| umts_disconnect_now | beliebig | Trigger Variable für manuelles Auflegen der UMTS Verbindung. Änderung des Wertes aktiviert den Vorgang. Nur bei ondemand = enabled | |
| umts_bands_2G<br>umts_bands_3G<br>umts_bands_4G | Liste | Mit Leerzeichen getrennte Liste von Bändern (siehe Status umts_supported_bands_json) oder ‚any‘ | |

## 1.8.2 3G/4G Status Abfragen

| Name | Parameter | Wert | Beschreibung |
|---|---|---|---|
| umts_multistat | - | Multiline Text | Umfangreicher Detailstatus des UMTS Moduls |
| umts_iccid | - | ICCID String | ICCID, eindeutiger Identifier der SIM Karte, „Error“ wenn keine Karte eingelegt ist. |
| umts_pinstate | - | String | READY wenn die PIN richtig gesetzt ist, „bad pin“ wenn sie falsch ist. |
| umts_signal | - | SNR | Empfangsstärke in dBm |
| umts_regstate | - | 0-5 | Status ID der Netzeinbuchung:<br>0: Nicht angemeldet, keine Netzsuche<br>1: Angemeldet am Heimatnetz<br>2: Netzsuche<br>3: Anmeldung nicht erlaubt<br>4: Unbekannt<br>5: Angemeldet im Fremdnetz (Roaming) |
| umts_localip | - | IP | IP Adresse des Gerätes falls bereit |
| umts_remoteip | - | IP | IP Adresse der Gegenstelle falls bereit |
| umts_operator | - | String | „0“ bei Fehler, ansonsten String des Operators, z.B. „vodafone.de“ |
| umts_stat | - | (connected \| standby – connect on demand \| not connected \| | Zustand der Verbindung, z.B. „connected“ wenn Verbunden. |

| | | | |
|---|---|---|---|
| | | connecting…) | |
| umts_serving_system | - | Json-Object | Bespiel: {"registration":"registered", "plmn_mcc":262, "plmn_mnc":1, "plmn_description":"Telekom.de", "roaming":false} |
| umts_supported_bands_json | - | Json Object | { "2G": { "G1900": 2097152, "G850": 524288, "G900": 256, "G1800": 128 }, "3G": { "B8": 562949953421312, "B5": 67108864, "B2": 8388608, "B1": 4194304 }, "4G": { "B20": 524288, "B8": 128, "B7": 64, "B3": 4, "B1": 1 } } G1900, B1, … sind gültige Bändernamen für die Config-Variablen umts_bands_{2,3,4}G, die zugewiesenen Werte sind für interne Zwecke. |

# 1.9 Webserver

## 1.9.1 Webserver Konfiguration

| Name | Wert | Beschreibung |
|---|---|---|
| service_https | (enabled\|disabled) | Falls disabled wird der HTTPS Zugang des Gerätes auf TCP Port443 nicht aktiviert. |
| service_http | (enabled\|disabled) | Falls disabled wird sowohl der HTTPS Zugang des Gerätes auf TCP Port443 als auch der HTTP Zugang auf TCP Port 80 nicht aktiviert.<br>(ab IRF2000 2.5.1, nur über API) |
| service_http_port | TCP Port | Per Default auf dem Wert „80“. Der HTTP Webserver kann hiermit auf einen anderen TCP Port verschoben werden. Der HTTPS TCP Port 443 bleibt davon unbeeinflusst.<br>(ab IRF2000 2.5.1, nur über API)) |

# 1.10 Spezielle Syntax für dynamische Tabellen

## 1.10.1 Allgemein

z.Zt. gibt es die folgenden dynamischen Tabellen, die also nicht aus Name/Wert-Paaren bestehen:

- Netzwerk-Gruppen
- Hardware-Gruppen
- Benutzerrechte
- Maschinendaten

Jede der Tabellen hat eine individuelle Anzahl von Werten pro Eintrag, d.h. Tabellenspalten.

### 1.10.1.1 Netzwerk-Gruppen

Tabellen-Name: ipgroups
constraint: die Kombination aus name und network muss einmalig sein

| Spaltenname | Wert | Beschreibung |
|---|---|---|
| name | Text, max 14 Zeichen | Der Name der Netzwerk-Guppe |
| network | IP/Maske | IP/Maske des Subnetzes, z.B. 192.168.1.0/24 |

### 1.10.1.2 Hardware-Gruppen

Tabellen-Name: macgroups
constraint: die Kombination aus name und hwaddr muss einmalig sein

| Spaltenname | Wert | Beschreibung |
|---|---|---|
| name | Text, max 14 Zeichen | Der Name der Netzwerk-Guppe |
| hwaddr | MAC-Adresse | Physikalische Hardware-Adresse, z.B. 00:10:20:45:67:89 |

### 1.10.1.3 Benutzerrechte

Tabellen-Name: permissions
constraint: die Kombination aus username und configid muss einmalig sein

| Spaltenname | Wert | Beschreibung |
|---|---|---|
| username | Text, max 14 Zeichen | Der Name der Netzwerk-Guppe |
| configid | (0\|1-9999\|[TABELLENNAME]) | Entweder<br>0: default-Wert für den Benutzer<br>1-9999: id der config DB variablen<br>TABELLENNAME: Name einer Tabelle, z.B. macgroups |

| | | |
|---|---|---|
| permission | (r\|w) | r: read, Leserecht w: write, Schreibrecht |

# 1.11 GNSS/GPS

## 1.11.1 Statusabfrage

| Name | Parameter 1 | Wert | Beschreibung |
|---|---|---|---|
| gnss_devices | | String | Liefert eine Liste von GNSS-Modems |
| gnss_tpv | | Json-Object | Liefert ein Json-Objekt mit<br>time: Zeit im ISO8601 Format<br>lat: Breitengrad<br>lon: Längengrad<br>alt: Höhe [m]<br>track: Geschwindigkeit [m/s] |
| gnss_sky | | Json-Array | Liefert ein Json-Array mit Objekten<br>in denen die empfangenen Satelliten gelistet sind.<br>PRN: Satellitenidentifikationsnummer |