www.vscom.de

viaVPN Daemon For Baltos

Edition: August 2017

Tel: +49 40 528 401 0
Fax: +49 40 528 401 99
Web: www.visionsystems.de
Support: service@visionsystems.de

The software described in this manual is furnished under a license agreement and may be used only in accordance with the terms of that agreement.

Copyright Notice

Trademarks

VScom is a registered trademark of Vision Systems GmbH. All other trademarks and brands are property of their rightful owners.

Disclaimer

Vision Systems reserves the right to make changes and improvements to its product without providing notice.

Vision Systems provides this document “as is”, without warranty of any kind, either expressed or implied, including, but not limited to, its particular purpose. Vision Systems reserves the right to make improvements and/or changes to this manual, or to the products and/or the programs described in this manual, at any time.

Information provided in this manual is intended to be accurate and reliable. However, Vision Systems assumes no responsibility for its use, or for any infringements on the rights of third parties that may result from its use.

This product might include unintentional technical or typographical errors. Changes are periodically made to the information herein to correct such errors, and these changes are incorporated into new editions of the publication.

Contents

List of Figures

1. Introduction

1.1. viaVPN Service

The system viaVPN provides Secure Remote Monitoring and Management via the Internet. In general there is a trade-off between ease-of-use and security, viaVPN optimizes this. The system is easy to install and use, but at the same time offers security given by strong encryption standards.

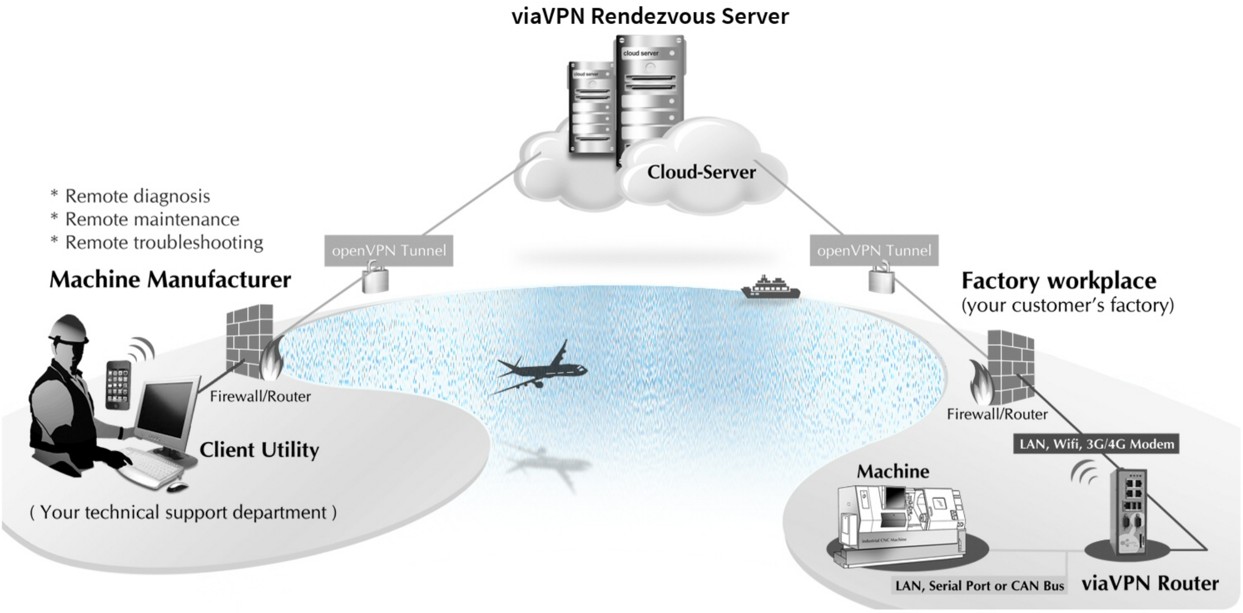


Figure 1: viaVPN Service

To perform all operations the viaVPN service uses four components. The central component is a so-called Rendezvous Server accessible in public Internet. Then there is a specialized Router system for viaVPN, and also a Client Utility software to start and stop service personnel connections to the Rendezvous Servers. Finally there is the Administration Web Interface on the Rendezvous Servers, used to manage privileges for access and connections.

The required connections to the Rendezvous Servers do not force any configuration changes in the company firewall. Further in most installations they do not need any configuration for the VPN-tunnel (on Router or PC) either, the default configuration is just fine.

1.2. viaVPN Daemon for Baltos

Section 1.1 describes the viaVPN system in general. In this case viaVPN router device just provides access to the LAN network using a VPN tunnel. viaVPN Daemon for Baltos (`vsopenvpnd`) enables first of all access to Baltos itself, i.e. you get direct access to your services running on Baltos like ssh, ftp etc. (see Figure 2 on page 5) It is of course possible to access network devices connected to the LAN port of the Baltos deivce. viaVPN service helps you to achieve this bypassing firewalls because both your workstation and Baltos connect to Rendezvous Server through standard HTTPS connection.

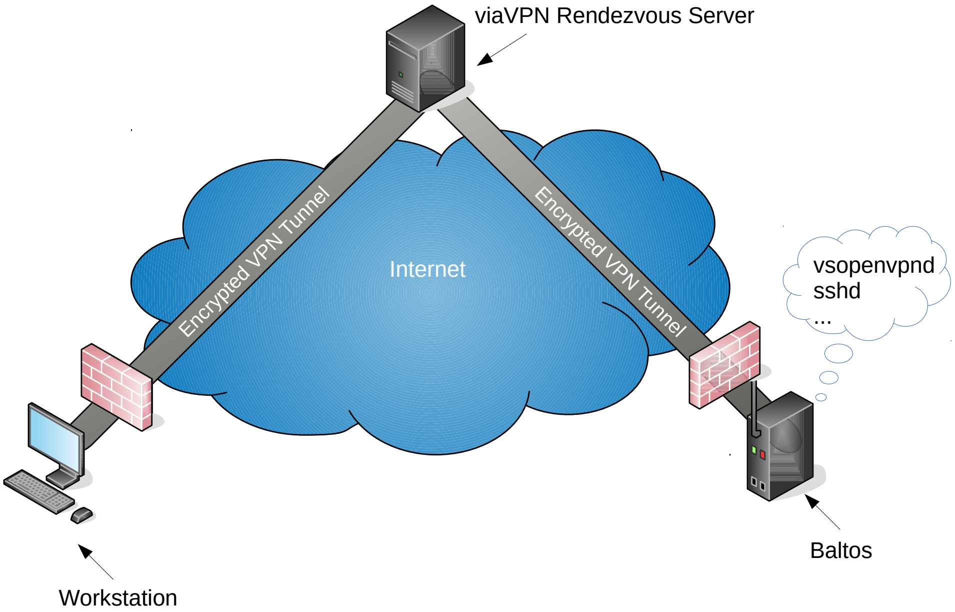


Figure 2: Baltos: viaVPN Connection Diagram

viaVPN service has following requirements:

1. viaVPN account
2. per device viaVPN data archive (OpenVPN certificates etc.)
3. Baltos device must have Internet access to be able to connect to the viaVPN Rendezvous Server

viaVPN account is the first thing you need to get before starting to work with viaVPN service. To create an account you'll need an AuthKey. This key will be supplied together with Baltos, when you order it with viaVPN option. It is either printed on the Baltos sticker or you can extract it from the `AuthKeys.html` file (see Section 2.2). With an AuthKey at hand proceed to the Section "Create an Account with viaVPN" of the viaVPN Manual: Administration Web Interface.

Now you're ready to install viaVPN daemon.

2. viaVPN Daemon Installation

2.1. Requirements

The Baltos must be running Debian 8 or 9 and use udev daemon for device and firmware management. This is the case for our own Debian images. Buildroot and Yocto distributions are supported too.

2.2. Getting Installtation Package

When ordering viaVPN support for already purchased devices the customer is to supply related serial numbers. In return the customer will get a file `viavpn-bundle.zip` containing following items:

- `viavpn_install.sh` - installation script for Debian
- `viavpn-sernum.zip` - a per device/serial number installation package providing required `*.deb` packages as also OpenVPN certificates
- `AuthKeys.html` - contains the registration code (AuthKey) and a link for each device allowing easy account creation and device registration

The `viavpn-bundle.zip` should be extracted to the root folder of customers USB mass storage device so that after mounting it to `/mnt` you'll have following file structure:

```
/mnt/AuthKeys.html
/mnt/viavpn-sernum.zip
/mnt/viavpn_install.sh
```

2.3. Deamon Installation in Debian

Insert prepared USB mass storage device into one of Baltos USB ports and invoke following commands:

1. `mount /dev/sda1 /mnt`
2. `/mnt/viavpn_install.sh`
3. wait till the whole installation is over. Can take some minutes to accomplish
4. perform a system reboot

After reboot `vsopenvpnd` will be running on startup (see Table 1 on page 13 for related LED blinking scheme).

2.4. Deamon Installation in Buildroot

In addition to our OnRISC Buildroot BSP we provide a special Buildroot BSP for viaVPN Daemon, that is also hosted on GitHub[1]. In order into integrate `vsopenvpnd` in your project follow instructions provided in README.md.

Having your system up and running you can now install OpenVPN certificates. Insert prepared USB mass storage device into one of Baltos USB ports and invoke following commands assuming that you perform these steps on a device with serial number 650100420:

1. `mount /dev/sda1 /mnt`
2. `cd /tmp`
3. `unzip /mnt/viavpn-650100420.zip`
4. `mkdir -p /opt/viavpn/data`

[1] https://github.com/visionsystemsgmbh/onrisc_br_viavpn

5. `unzip /tmp/viavpn-data.zip -d /opt/viavpn/data/`

To check the installation make sure Baltos has correct date/time settings and has Internet access. Invoke:

```
vsopenvpnd -P /opt/viavpn/data
```

You'll get following output showing a successful connection:

```
# vsopenvpnd -P /opt/viavpn/data/
VScom OpenVPN Daemon (1.0.0) started
vsopenvpnd: init: RDV server list:
vsopenvpnd: init: master.viavpn.com
vsopenvpnd: init: slave.viavpn.com
vsopenvpnd: web: getting VPN connection data from https://master.viavpn.com
Starting XML-RPC server 1.0.0
vsopenvpnd: web: connecting to VPN 23.239.9.207:443
vsopenvpnd: init: try second resolv
vsopenvpnd: init: DNS server: 213.209.99.202
vsopenvpnd: mgmt: connected
vsopenvpnd: vpn: state - CONNECTED
invoke up
vsopenvpnd: udp control: started
```

The Section Daemon Configuration describes, how to configure `vsopenvpnd` and automate tasks on connection establishment and termination.

2.5. Deamon Installation in Yocto

Our `meta-baltos` layer already provides a recipe for `vsopenvpnd` package. Get familiar with our Yocto BSP described in the OnRISC User Manual[2]. After doing this you'll first of all need to select `vsopenvpnd` package. This can be achieved via adding `vsopenvpnd` right after `libonrisc` in the `conf/local.conf` file:

```
IMAGE_INSTALL_append = " libonrisc vsopenvpnd"
```

Now invoke `bitbake core-image-base` and you'll get a rootfs providing `vsopenpnvd`. The viaVPN data installation steps are the same as described in Section Deamon Installation in Buildroot.

3. Establishing the First viaVPN Connection

In previous Sections you've created a viaVPN account, added and paired your device and `vsopenvpnd` daemon is up und running. Now it's time to install the viaVPN Client Utility and connect to the Baltos device. Download viaVPN Manual: viaVPN Client Utility and perform actions described in Section "Install viaVPN Client Utility".

After installing and logging in you should see "Baltos viaVPN Daemon" among other devices provided you've input this string into the "Name" field during the device adding (see Figure 3 on page 11). Select this device and click on the "Connect" button. Upon a successful connection you should

[2] ftp://ftp.visionsystems.de/pub/multiio/OnRISC/Baltos/OnRISC_User_Manual.pdf

see that the color of “Baltos viaVPN Daemon” has changed to light green (see Figure 4 on page 11).

3.1. Configuring DHCP Server

Now you have a TAP interface on your Windows host, but it has no IP address. By default viaVPN Client Utility will try to aquire its IP address from Batlos device via DHCP (refer to Section 4.2 for other options). So click on “Disconnect” botton and prepare to configure DHCP server on Baltos.

dnsmasq[3] is a very popular DHCP/TFTP server for embedded systems. It is already available if you use Buildroot BSP and on Debian you can easliy install invoking following command:

```
apt install dnsmasq
```

Create a file /etc/dnsmasq.conf with following content:

```
interface=tap0
dhcp-range=10.0.10.50,10.0.10.150,10m
dhcp-option=3
dhcp-option=6
```

Two fisrt lines configure DHCP service on tap0 interface assigning IP addresses from 10.0.10.50 to 10.0.10.150. You can change this range according to your needs. DHCP options are needed to disable setting default gateway to TAP interface in Windows.

3.2. Configuring vsopenvpnd Hooks

As tap0 interface is not availlable on startup we need to start dnsmasq right after establishing a VPN connection. This is where vsopenvpnd hooks come in play. Open /opt/viavpn/scripts/up and fill it with following content according to your Linux distribution.

For Buildroot you’ll need following commands:

```
#!/bin/sh

ip addr add 10.0.10.1/24 dev tap0
/etc/init.d/S80dnsmasq restart
```

For Debian you’ll need following commands:

```
#!/bin/sh

ip addr add 10.0.10.1/24 dev tap0
systemctl restart dnsmasq
```

This script configures IP address for tap0 interface and restarts dnsmasq. And we also need a hook when closing the VPN connection (/opt/viavpn/scripts/down).

For Buildroot you’ll need following commands:

[3] http://www.thekelleys.org.uk/dnsmasq/doc.html

```
#!/bin/sh

/etc/init.d/S80dnsmasq stop
```

For Debian you'll need following commands:

```
#!/bin/sh

systemctl stop dnsmasq
```

Now we will create /opt/viavpn/viavpn.conf in order to store all needed configuration:

```
daemon
path /opt/viavpn/data
up /opt/viavpn/scripts/up
down /opt/viavpn/scripts/down
```

Invoke and vsopenvpnd will and go to background at once as we've configured it to run as a daemon:

```
vsopenvpnd -c /opt/viavpn/viavpn.conf
```

You can see the system messages using tail utility:

```
tail -f /var/log/messages
```

3.3. Checking TAP Device Settings in Windows

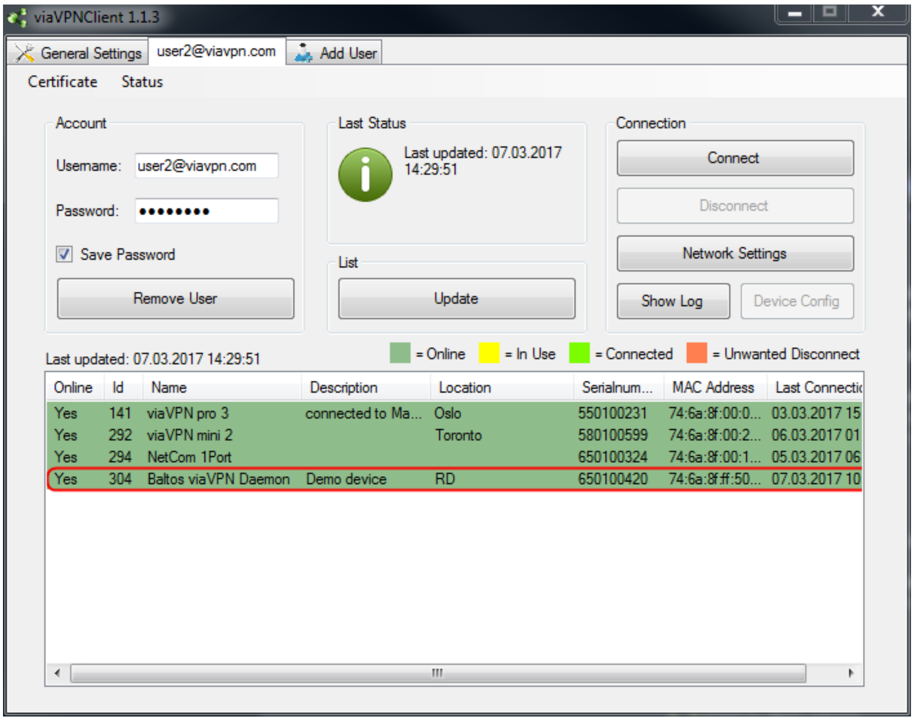
Invoking ipconfig /all in Windows will show similar output for your TAP interface:

```
Ethernet adapter Local Area Connection 4:
Connection-specific DNS Suffix .  :
Description . . . . . . . . . . . . :  TAP-Windows Adapter V9
Physical Address. . . . . . . . . . :  00-FF-DF-AC-19-D7
DHCP Enabled. . . . . . . . . . . . :  Yes
Autoconfiguration Enabled . . . . :  Yes
Link-local IPv6 Address . . . . . :  fe80::71f5:3f23:23c:eab0%26(Preferred)
IPv4 Address. . . . . . . . . . . . :  10.0.10.98(Preferred)
Subnet Mask . . . . . . . . . . . . :  255.255.255.0
Lease Obtained. . . . . . . . . . . :  Monday, 7.  March 2017 11:13:46
Lease Expires . . . . . . . . . . . :  Monday, 7.  March 2017 11:23:46
Default Gateway . . . . . . . . . :
DHCP Server . . . . . . . . . . . . :  10.0.10.1
DHCPv6 IAID . . . . . . . . . . . . :  687931359
DHCPv6 Client DUID. . . . . . . . :  00-01-00-01-17-EC-D8-E3-90-2B-34-36-5F-B0
DNS Servers . . . . . . . . . . . . :  fec0:0:0:ffff::1%1
fec0:0:0:ffff::2%1
fec0:0:0:ffff::3%1
NetBIOS over Tcpip. . . . . . . . :  Enabled
```

As one can see your TAP interface got an IP address from dnsmasq range 10.0.10.98 and Default Gateway entry is empty. You can now ping the Baltos devices on its IP address 10.0.10.1:

```
Pinging 10.0.10.1 with 32 bytes of data:
Reply from 10.0.10.1:  bytes=32 time=68ms TTL=64
Reply from 10.0.10.1:  bytes=32 time=34ms TTL=64
Reply from 10.0.10.1:  bytes=32 time=34ms TTL=64
Reply from 10.0.10.1:  bytes=32 time=34ms TTL=64
Ping statistics for 10.0.10.1:
Packets:  Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 34ms, Maximum = 68ms, Average = 42ms
```

From this point you can connect to your services running on Baltos like SSH, FTP server etc.



Username: user2@viavpn.com

Password: ••••••••

☑ Save Password

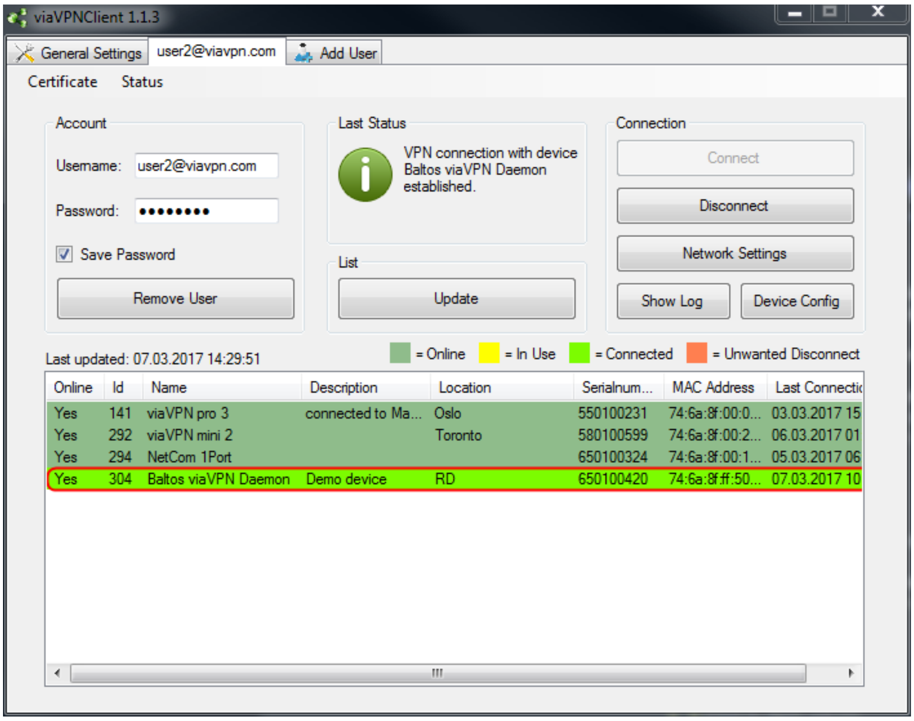
Last updated: 07.03.2017 14:29:51

Last updated: 07.03.2017 14:29:51

🟩 = Online 🟨 = In Use 🟩 = Connected 🟧 = Unwanted Disconnect

Online	Id	Name	Description	Location	Serialnum...	MAC Address	Last Connectio
Yes	141	viaVPN pro 3	connected to Ma...	Oslo	550100231	74:6a:8f:00:0...	03.03.2017 15
Yes	292	viaVPN mini 2		Toronto	580100599	74:6a:8f:00:2...	06.03.2017 01
Yes	294	NetCom 1Port			650100324	74:6a:8f:00:1...	05.03.2017 06
Yes	304	Baltos viaVPN Daemon	Demo device	RD	650100420	74:6a:8f:ff:50...	07.03.2017 10

Figure 3: The List of Connected Devices



Username: user2@viavpn.com

Password: ••••••••

☑ Save Password

VPN connection with device Baltos viaVPN Daemon established.

Last updated: 07.03.2017 14:29:51

🟩 = Online 🟨 = In Use 🟩 = Connected 🟧 = Unwanted Disconnect

Online	Id	Name	Description	Location	Serialnum...	MAC Address	Last Connectio
Yes	141	viaVPN pro 3	connected to Ma...	Oslo	550100231	74:6a:8f:00:0...	03.03.2017 15
Yes	292	viaVPN mini 2		Toronto	580100599	74:6a:8f:00:2...	06.03.2017 01
Yes	294	NetCom 1Port			650100324	74:6a:8f:00:1...	05.03.2017 06
Yes	304	Baltos viaVPN Daemon	Demo device	RD	650100420	74:6a:8f:ff:50...	07.03.2017 10

Figure 4: Successful Connection

4. Daemon Configuration

By default all configuration files will be installed to `/opt/viavpn`, but this location can be freely configured. viaVPN Daemon uses following configuration files:

- `viavpn.conf` - configuration file for `vsopenvpnd`
- `scripts/` - contains bash files that will be invoked at different connection states of `vsopenvpnd`
- `data/` - contains info for encryption and authencation with viaVPN service

4.1. viavpn.conf

Below you'll find `viavpn.conf` fragments with related explanations.

4.1.1. Internal Data Path

```
# Configuration file for the viaVPN Daemon

# Path to viaVPN data
path /opt/viavpn/data
```

`path` defines where the encryption and authentication data is found.

4.1.2. Daemon Mode

```
# Start as Daemon
daemon
```

`daemon` means that `vsopenvpnd` will fork to run in the background when started.

4.1.3. LED Configuration

```
#Status LED
#led app
```

`led` defines what LED will be used for visual state signaling (refer to Section 4.3). Possible values are: `pwr`, `wln` and `app` (default).

4.1.4. Proxy Settings

```
# Proxy settings

#host 172.16.0.1
#port 8080
#user surfer
#password secret
```

If a proxy is needed for Internet access, one can set the appropriate settings here.

4.1.5. Scripts

```
# Scripts

up /opt/viavpn/scripts/up
on /opt/viavpn/scripts/on
off /opt/viavpn/scripts/off
down /opt/viavpn/scripts/down
```

Connection states:

- `up` - This state is entered when your device is connected to the rendezvous server and the tap0 interface is present.
- `on` - This state is entered when a User connects to the device via Internet using the viaVPN Client Utility. In this state data can flow through the VPN tunnel.
- `off` - When a User disconnects this state is entered
- `down` - This state is entered when your device lost the connection to the rendezvous server

Each of these states can trigger actions to be performed on the device. Four example bash scripts are defined and can be found under `/opt/viavpn/scripts.`

4.2. IP Address Resolution / DHCP

The viaVPN Client Utility has three options for IP address assignment:

- Automatically over DHCP - to use this option with your devices a DHCP server has to run to assign an IP address to the Client. Example: Client 10.0.10.18/24 <—> 10.0.10.1/24 tap0 on device. This option is recommended
- Static IP Address - In this case a static IP has be set in the viaVPN Client Utility which may be an inconvenience for the User
- viaVPN Subnet - This option sets a static IP (172.31.31.100/24), but it should be avoided if possible

4.3. LED Blinking Scheme

`vsopenvpnd` shows different operating states via green LED named APP. See table below:

State	**On time (seconds)**	**Off time (seconds)**
Connecting to Rendezvous Server	1.5	1.5
Connection to Rendezvous Server established	permanent	0
viaVPN Client Utility connection	2	1
Not able to connect to Rendezvous Server	0.5	2.5
viaVPN configuration missing	0.1	0.1

Table 1: LED Blinking Scheme

A. History

Decmber 2016 Release viaVPN Daemon for Baltos

August 2017 Add Yocto installation instructions