
Installation EDOMI auf einem apu2-System (CentOS 6.5) v1.0

Diese Anleitung ist veraltet! Bitte verwende die neuere Version 2.0 mit CentOS 7.6.

Diese Anleitung beschreibt das Vorbereiten und Installieren von EDOMI auf einem apu2-System. Die Anleitung richtet sich an Linux-Anfänger, die einfach EDOMI lauffähig machen wollen.

Disclaimer: Dies ist keine offizielle Anleitung vom Entwickler von EDOMI. Die Anleitung wurde von keinem Linux-Profi geschrieben. Die Installation geschieht auf eigene Verantwortung. Die Installation wurde jedoch mehrfach erfolgreich getestet.

Für die nachfolgend gezeigten Schritte wurden folgende Komponenten verwendet:

- [apu2c4](#) Systemboard (AMD GX-412TC 64bit CPU, 4 GB RAM, 3x Intel LAN, 2x USB3, lüfterlos, kein Video-Ausgang!)
- Externes [Netzteil](#) 12V, 2A
- [Aluminium-Gehäuse](#), B:168 mm x T:157 mm x H:30 mm
- Kingston SSDNow mS200 - 60GB, mSATA
- [USB Nullmodem RS232 Adapter Kabel](#) (unter Windows 7 eingesetzt)
- apu2 [BIOS](#) v4.0.7
- [TinyCoreLinux](#) (für BIOS-Update)
- [CentOS 6.5 minimal ISO](#) (keine neuere Version verwenden! EDOMI ist für v6.5 ausgelegt.)
- [EDOMI v1.49](#)*
- [ISO2USB](#) (Achtung: Es gibt mehrere Tools mit diesem Namen. Link verwenden!)
- [Putty](#) (Windows)
- 4 GB grosser USB-Stick

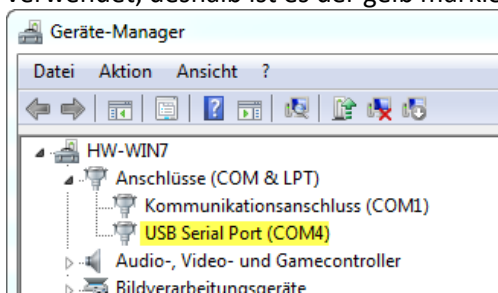
***Hinweis:** EDOMI wird häufig aktualisiert. Bitte die jeweils aktuellste Version verwenden. Die hier gezeigten Screenshots können entsprechend abweichen.

Vorbereiten serielle Verbindung mit apu2

Da das apu2-System kein Video-Ausgang besitzt, kann es nur via serieller Schnittstelle installiert werden! (Später, wenn das OS erfolgreich installiert und die Netzwerkeinstellungen gesetzt wurden, kann über SSH oder Webkonsole zugegriffen werden. Die serielle Schnittstelle wird dann nicht mehr benötigt.)

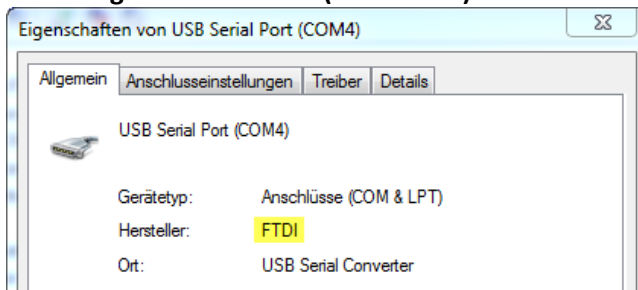
Es muss ein serielles **Nullmodem**-Kabel verwendet werden. Besitzt der Computer keine serielle Schnittstelle (ist häufig der Fall bei heutigen Geräten), so muss ein USB-Adapter verwendet werden. Hier unbedingt darauf achten, dass es sich um ein Nullmodem-Adapter handelt und dass das serielle Kabel ein DB9-Stecker (female) besitzt.

1. Nullmodem-Kabel am Computer anschliessen. (abhängig vom Betriebssystem und USB-Adapter-Modell müssen ggf. noch Treiber installiert werden.)
2. Im Windows Gerätemanager prüfen, ob das Nullmodem-Kabel angezeigt wird. Eine serielle Schnittstelle wird immer mit COM1, COM2, COM3 etc. bezeichnet. In meinem Beispiel wurde ein USB-Adapter verwendet, deshalb ist es der gelb markierte Eintrag:

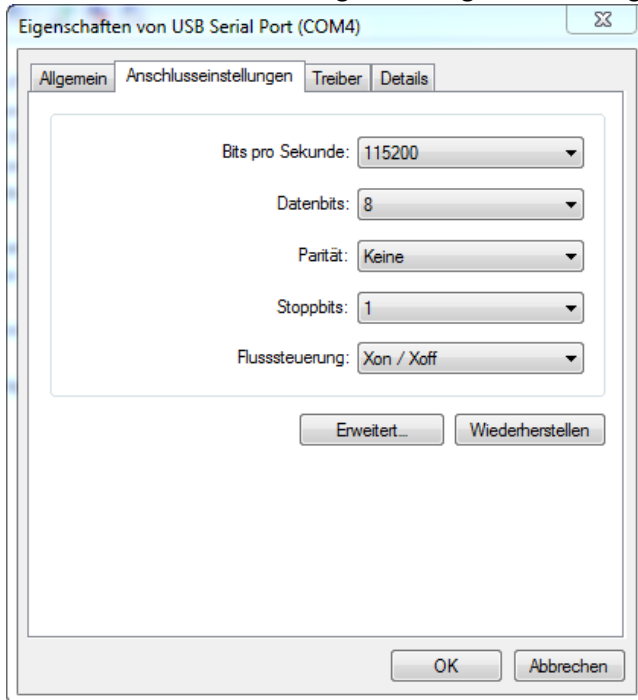


3. Auf den Eintrag doppelklicken und anhand des Herstellers prüfen, ob man auch wirklich den richtigen Adapter ausgewählt hat. (In meinem Fall handelt es sich um ein Produkt mit einem FTDI-Chip, somit ist dies der korrekte Eintrag)

Unbedingt den COM-Port (hier COM4) merken!

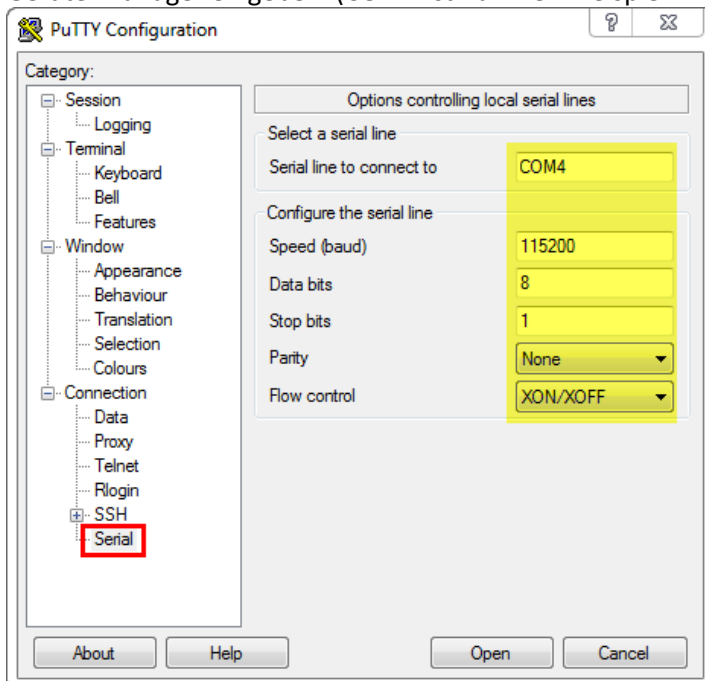


4. Bei den Anschlusseinstellungen die folgenden Einträge eingeben:



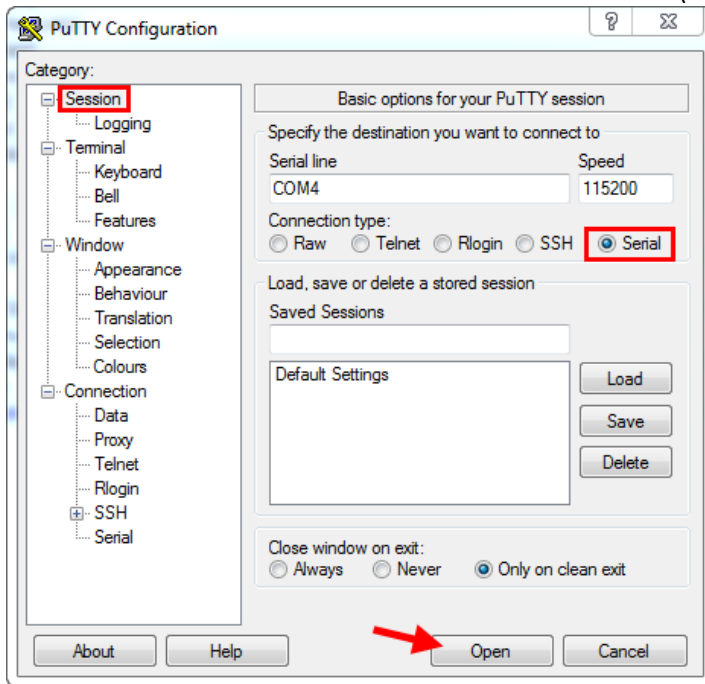
(Der Geräte-Manager kann nun wieder beendet werden.)

5. Das serielle Kabel nun mit dem apu2 verbinden und das Stromnetzteil anschliessen (das apu2-System hat kein Power-Knopf)
6. Das Programm **PuTTY** starten. Ganz unten bei **Connection > Serial** die gleichen Werte wie vorhin beim Geräte-Manager eingeben. (COM4 ist nur mein Beispiel. Muss ggf. angepasst werden.)

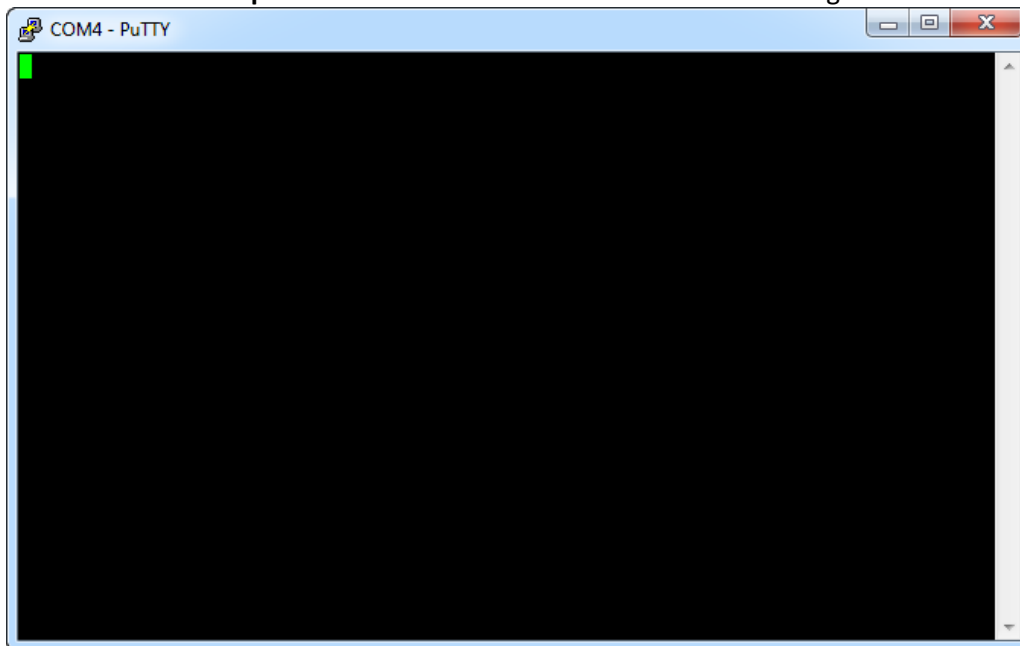


Achtung: Diese Werte werden beim Beenden von PuTTY wieder zurückgesetzt!

7. Zuerst auf **Session** und dann rechts auf **Serial** klicken. (Dieser Schritt wird oft vergessen!)



8. Nach Drücken von **Open** sollte ein leeres Konsolenfenster mit einem grünen Cursor oben links erscheinen.



Auch wenn dies nicht sehr spektakulär aussieht, so hast du nun erfolgreich eine serielle Verbindung mit dem apu2-System aufgebaut. (drückt man hier die RETURN-Taste, können ggf. weitere Ausgaben angezeigt werden.)

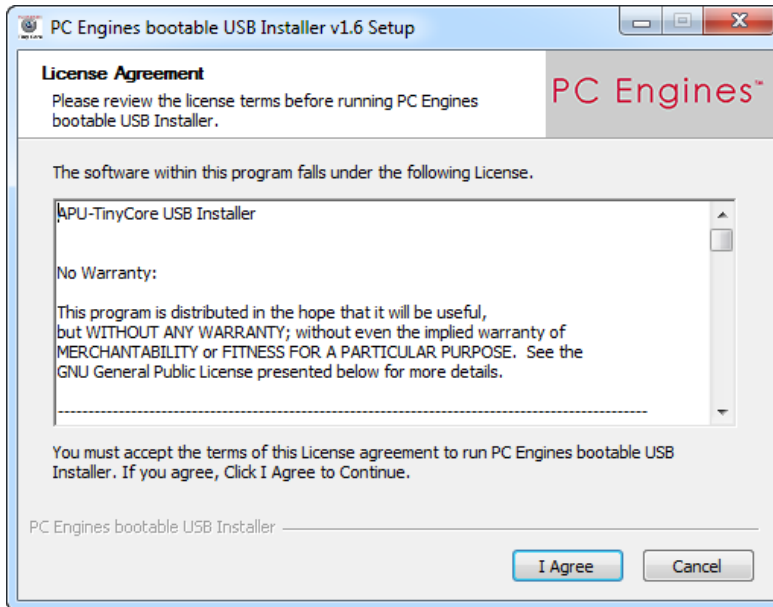
Erscheinen Putty-Fehlermeldungen (nicht in der Konsole!), so stimmen die seriellen Einstellungen nicht oder das Kabel ist falsch angeschlossen. Bitte nochmals alles prüfen.

Hinweis: Beim Einschalten/Starten eines apu2-Systems, kann mit der F10-Taste das BIOS aufgerufen werden. Da dies sehr schnell geschieht, sollte VOR dem Einschalten Putty bereits gestartet sein. Am besten man baut, wie oben beschrieben, eine serielle Verbindung auf und steckt anschliessend kurz das Netzteil (auf der 220V-Seite, nicht am apu2!) aus und ein. Anschliessend werden die apu2-Textanzeigen sofort im Konsolenfenster angezeigt und man kann rechtzeitig darauf reagieren.

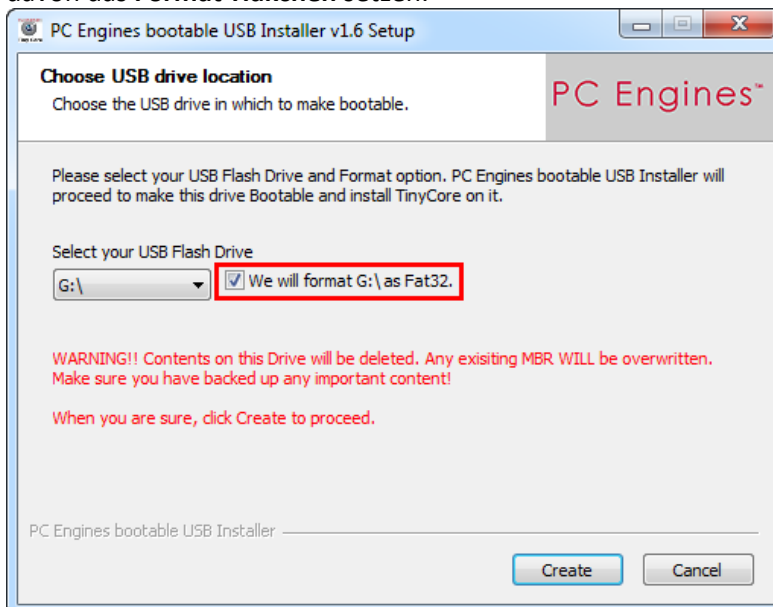
BIOS-Update des apu2 (optional)

Bei einem neu gekauften apu2 ist es sinnvoll, wenn man zuerst ein BIOS/Firmware-Update durchführt.

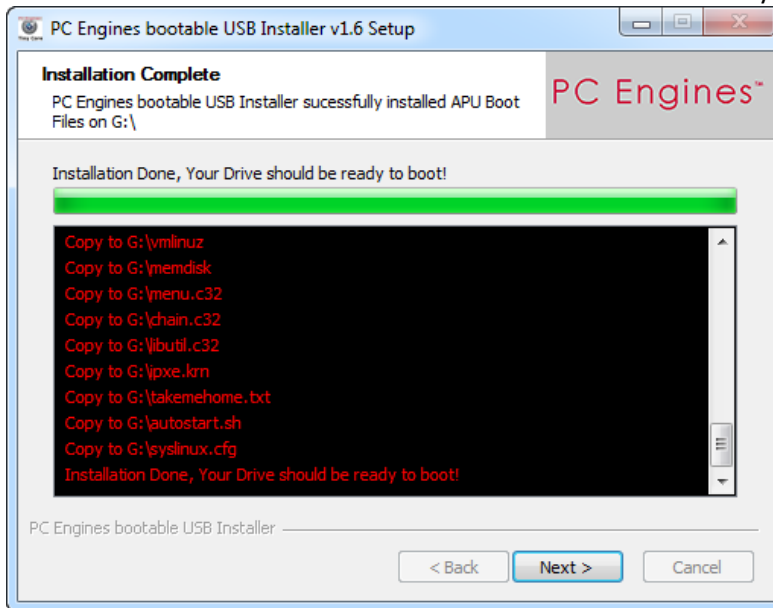
1. Einen 4 GB USB-Stick mit FAT32 formatieren (wird zwar später nochmals formatiert, aber so weiss man, dass der Stick in Ordnung ist.)
2. Das Tool **apu-bootable-usb-installer_v1.6.exe** (siehe Link TinyCoreLinux zu Beginn der Anleitung) als Administrator ausführen.



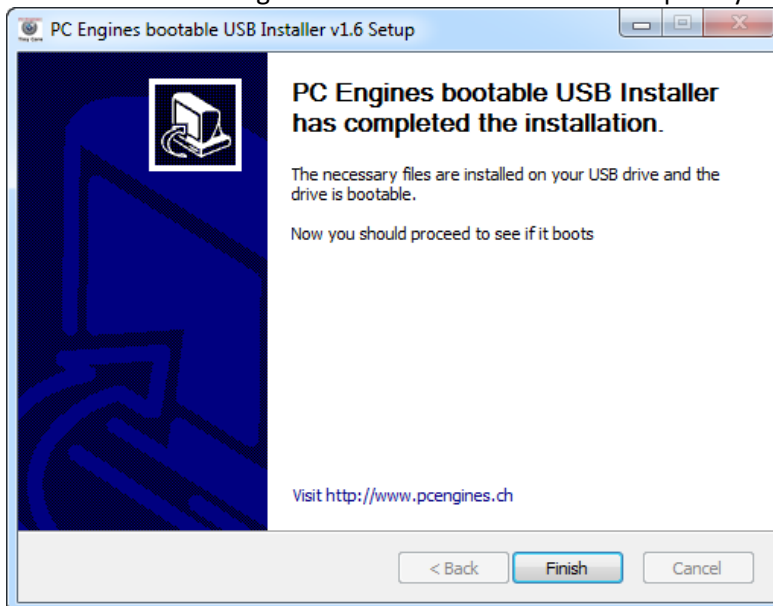
3. Den entsprechenden Laufwerksbuchstaben des vorher formatierten USB-Sticks auswählen und rechts davon das **Format-Häkchen** setzen.



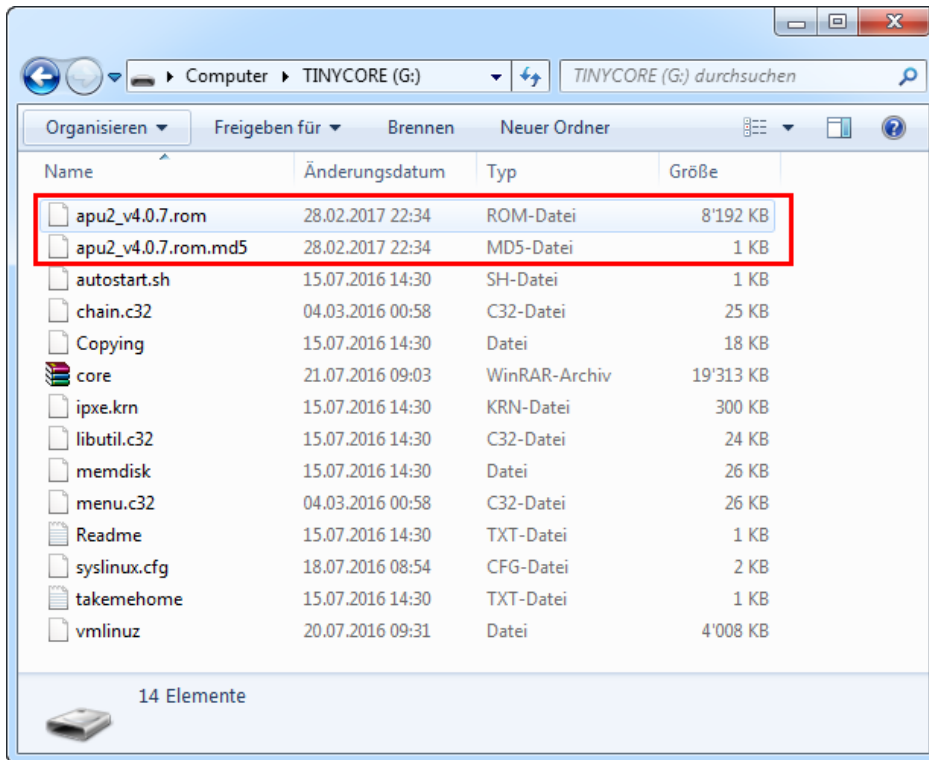
4. Mit **Create** wird der Stick nochmals formatiert und mit dem TinyCoreLinux beschrieben.



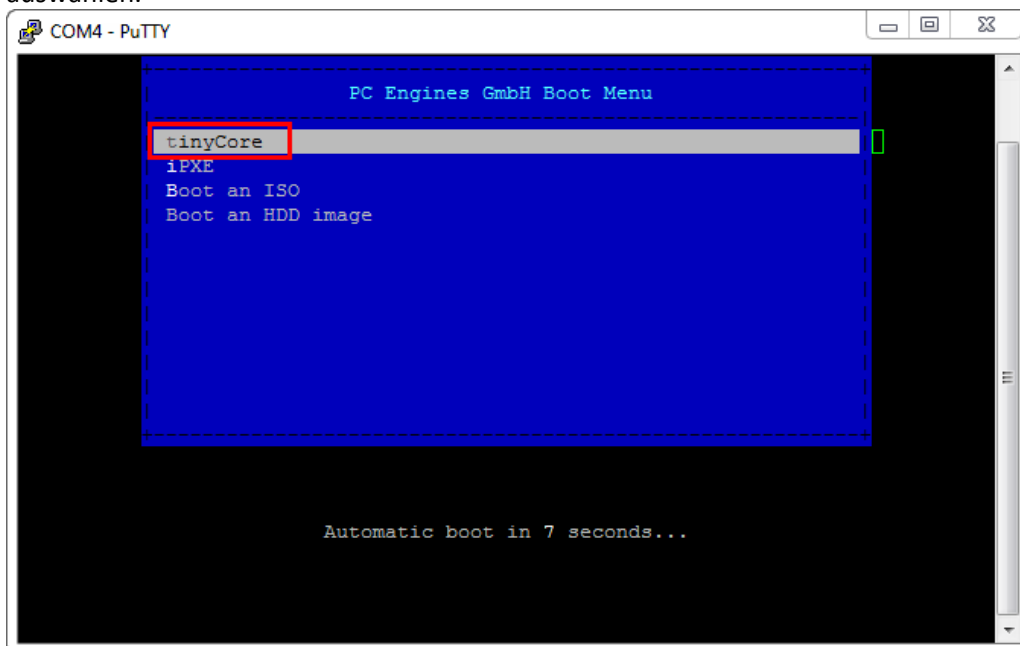
Der Stick ist nun fertig und kann zum Booten auf dem apu2-System verwendet werden.



5. Bevor der Stick ausgeworfen wird, muss noch das aktuelle BIOS darauf kopiert werden. In diesem Beispiel wurde das 4.0.7 ROM verwendet (die md5-Datei ist glaub nicht zwingend notwendig):



- Nun der vorbereitete USB-Stick in das apu2-System stecken und bei aktiver Putty-Verbindung das Gerät einschalten.
- Nach einer kurzen Zeit sollte das Bootmenü des Stick erscheinen. Hier den ersten Eintrag **tinyCore** auswählen.



- tinyCoreLinux wird nun gestartet und es erscheint ein Command Prompt. Zudem wird bereits der Befehl für das Firmware-Update angegeben. Bevor wir aber den Befehl ausführen können, müssen wir noch den USB-Stick mounten. Ansonsten können wir nicht auf die vorher kopierte BIOS-ROM-Datei zugreifen.

```

COM4 - PuTTY
Skipping DHCP broadcast/network detection as requested on boot commandline.
Setting hostname to box Done.

  PC Engines

TinyCore 7.2 www.tinycorelinux.com

login[538]: root login on 'ttyS0'
searching for home directory ...
FAT partition not mounted yet, retrying since 0 ssd 4:0:0:0: [sdb] Asking for ca
che data failed

Welcome to TinyCore 6.4 running on APU1/APU2
To update the BIOS type "flashrom -w apu_xyz.rom -p internal"
You can find GPIO examples in /root/gpio_examples

[+54.6 C][root@box:~]$

```

9. Mit dem Befehl `fdisk -l` können wir unsere Disks anzeigen. Der USB-Stick ist anhand der Grösse (hier 3.8GB) und des Typ FAT32 zu erkennen. Wichtig ist der Device-Name (hier `/dev/sdb1`).

```

COM4 - PuTTY
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disk /dev/sda: 55.9 GiB, 60022480896 bytes, 117231408 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x00035249

Device      Boot  Start      End  Sectors  Size Id Type
/dev/sda1  *           2048   1026047   1024000   500M 83 Linux
/dev/sda2           1026048 117229567 116203520 55.4G 8e Linux LVM

Disk /dev/sdb: 3.8 GiB, 4043308544 bytes, 7897087 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x00000000

Device      Boot  Start      End  Sectors  Size Id Type
/dev/sdb1  *           128   7896959   7896832   3.8G  b W95 FAT32

[+51.5 C][root@box:~]$

```

10. Nun können wir den USB-Stick mit dem Befehl `mount /dev/sdb1 /media/flash_drive_sdb` mounten. Ein anschliessender `ls /media/flash_drive_sdb` Befehl zeigt uns den Inhalt des Sticks an. Wir sollten damit auch die BIOS-Datei sehen.


```

COM4 - PuTTY
Device      Boot  Start      End  Sectors  Size Id Type
/dev/sda1  *      2048      1026047  1024000  500M 83 Linux
/dev/sda2      1026048  117229567  116203520  55.4G 8e Linux LVM

Disk /dev/sdb: 3.8 GiB, 4043308544 bytes, 7897087 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x00000000

Device      Boot  Start      End  Sectors  Size Id Type
/dev/sdb1  *      128 7896959 7896832  3.8G  b W95 FAT32

[+51.5 C][root@box:~]$ mount /dev/sdb1 /media/flash_drive_sdb
[+51.5 C][root@box:~]$ ls /media/flash_drive_sdb/
Copying          core.gz          menu.c32
Readme.txt       ipxe.krn        syslinux.cfg
apu2_v4.0.7.rom  ldlinux.c32     takemehome.txt
apu2_v4.0.7.rom.md5 ldlinux.sys     vmlinuz
autostart.sh     libutil.c32
chain.c32        memdisk
[+51.4 C][root@box:~]$

```

11. Da wir nun den Stick gemountet haben, können wir den eigentlichen Firmware-Update starten. Dazu den folgenden Befehl eintippen:
flashrom -w /media/flash_drive_sdb/apu2_v4.0.7.rom -p internal (der Name der ROM-Datei muss ggf. angepasst werden.)

```

COM4 - PuTTY
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x00035249

Device      Boot  Start      End  Sectors  Size Id Type
/dev/sda1  *      2048      1026047  1024000  500M 83 Linux
/dev/sda2      1026048  117229567  116203520  55.4G 8e Linux LVM

Disk /dev/sdb: 3.8 GiB, 4043308544 bytes, 7897087 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x00000000

Device      Boot  Start      End  Sectors  Size Id Type
/dev/sdb1  *      128 7896959 7896832  3.8G  b W95 FAT32

[+51.5 C][root@box:~]$ mount /dev/sdb1 /media/flash_drive_sdb
[+51.5 C][root@box:~]$ ls /media/flash_drive_sdb/
Copying          core.gz          menu.c32
Readme.txt       ipxe.krn        syslinux.cfg
apu2_v4.0.7.rom  ldlinux.c32     takemehome.txt
apu2_v4.0.7.rom.md5 ldlinux.sys     vmlinuz
autostart.sh     libutil.c32
chain.c32        memdisk
[+51.4 C][root@box:~]$ flashrom -w /media/flash_drive_sdb/apu2_v4.0.7.rom -p internal

```

12. Der Update-Prozess dauert nicht sehr lange. Da in meinem Fall das BIOS bereits die gleiche Version wie auf dem Stick aufweist, wurde kein Update mehr ausgeführt. Erscheint wieder der Command Prompt, sollte der Update erfolgreich installiert sein (Meldungen durchlesen!).

```
COM4 - PuTTY
Device      Boot Start      End Sectors  Size Id Type
/dev/sdb1   *          128 7896959 7896832  3.8G b W95 FAT32

[+51.5 C][root@box:~]$ mount /dev/sdb1 /media/flash_drive_sdb
[+51.5 C][root@box:~]$ ls /media/flash_drive_sdb/
Copying          core.gz          menu.c32
Readme.txt       ipxe.krn        syslinux.cfg
apu2_v4.0.7.rom  ldlinux.c32     takemehome.txt
apu2_v4.0.7.rom.md5 ldlinux.sys     vmlinuz
autostart.sh     libutil.c32
chain.c32        memdisk

[+51.4 C][root@box:~]$ flashrom -w /media/flash_drive_sdb/apu2_v4.0.7.rom -p internal
flashrom v0.9.8-r1888 on Linux 4.2.9-tinycore (i686)
flashrom is free software, get the source code at http://www.flashrom.org

Calibrating delay loop... OK.
Error accessing high tables, 0x100000 bytes at 0xdffae000
/dev/mem mmap failed: Resource temporarily unavailable
Failed getting access to coreboot high tables.
Found chipset "AMD FCH".
Enabling flash write... OK.
Found Winbond flash chip "W25Q64.V" (8192 kB, SPI) mapped at physical address 0xff800000.
Reading old flash chip contents... done.
Erasing and writing flash chip...
Warning: Chip content is identical to the requested image.
Erase/write done.
[+55.0 C][root@box:~]$
```

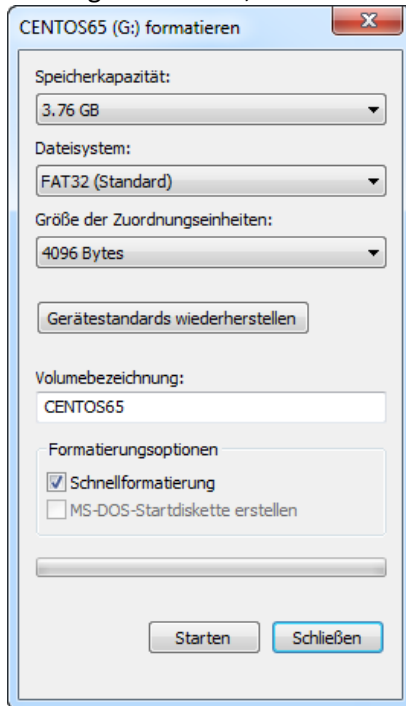
Der USB-Stick kann nun wieder für andere Zwecke verwendet werden (vorher am besten formatieren).

CentOS 6.5 minimal USB-Stick vorbereiten

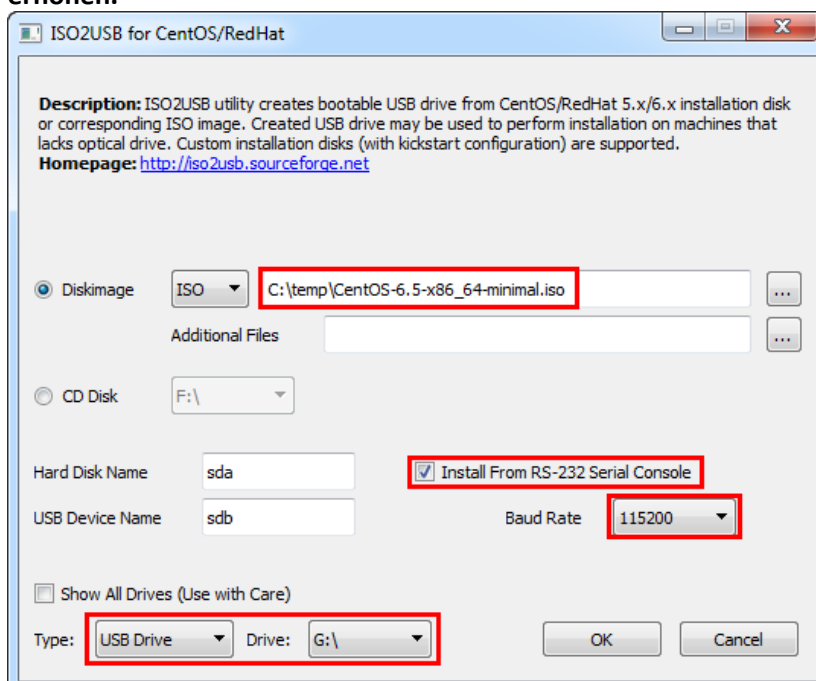
Bevor wir EDOMI installieren können, müssen wir zuerst das benötigte LINUX-Betriebssystem installieren. Im Falle von EDOMI ist dies **CentOS 6.5 minimal** (und bitte nur dieses verwenden!).

Auch hier beginnen wir mit dem Vorbereiten eines entsprechenden, bootfähigen USB-Stick.

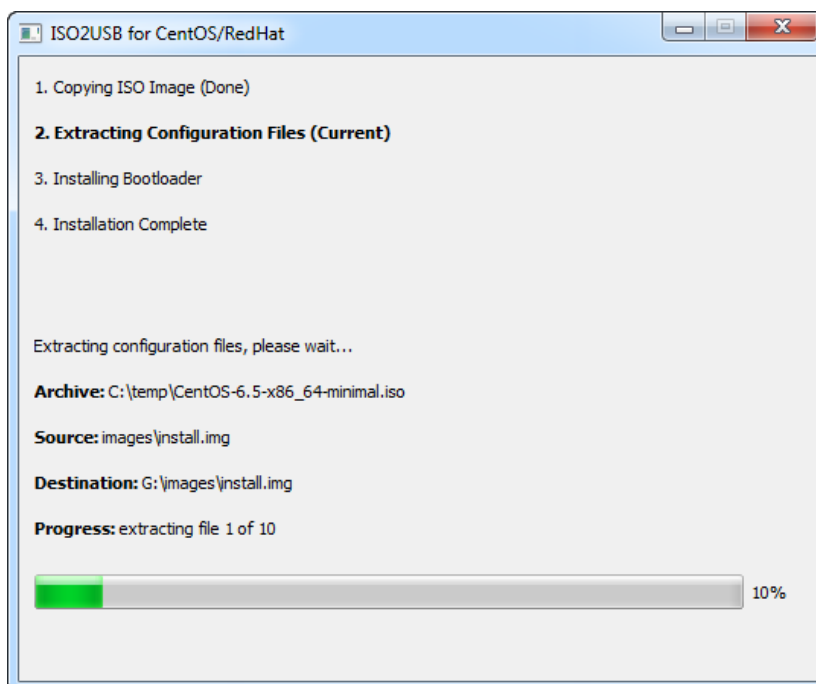
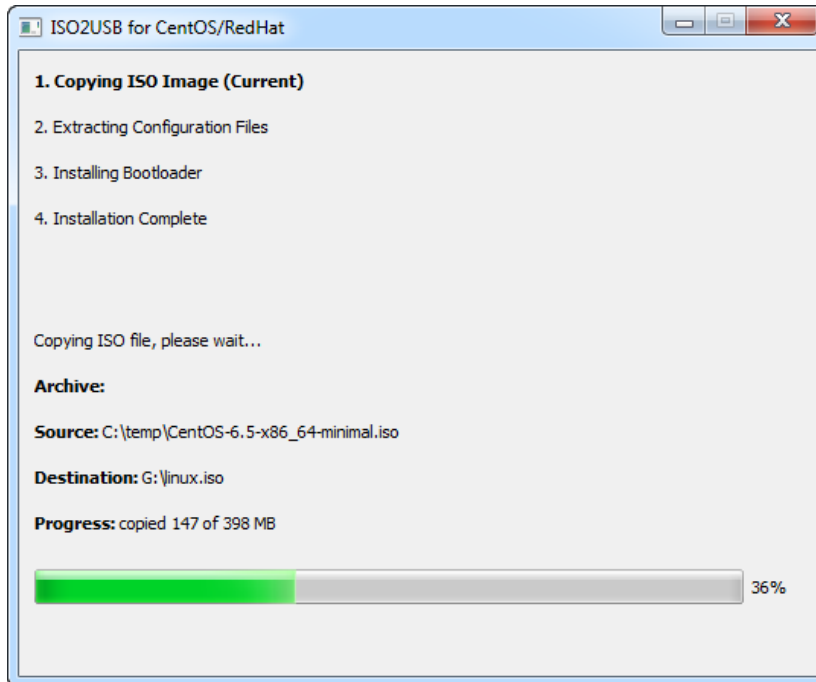
1. Wir beginnen damit, dass wir den USB-Stick wieder mit **FAT32** formatieren:



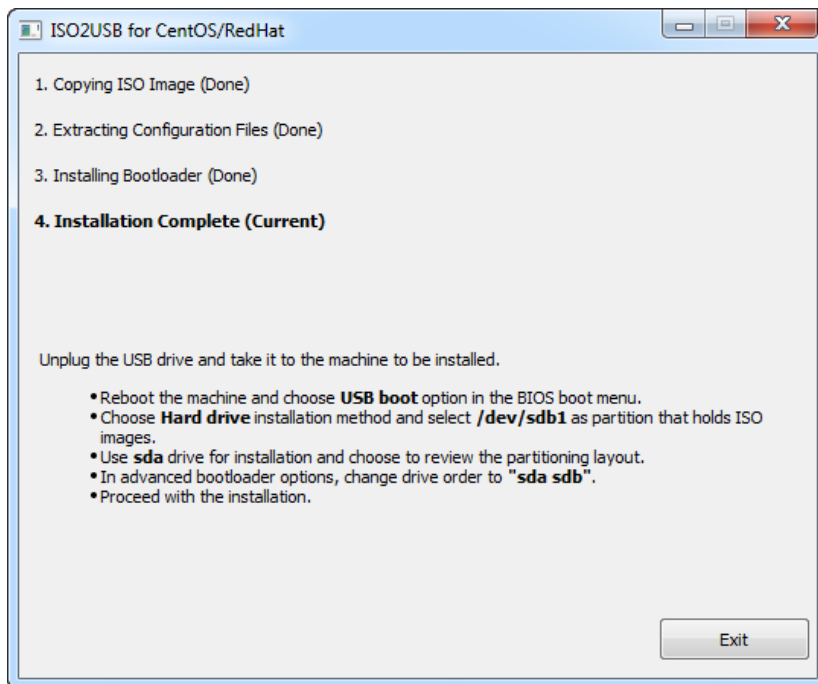
2. Wir starten nun das Tool **ISO2USB** mit Administratorenrechten und wählen als Diskimage das vorher downgeladete **CentOS-6.5-x86_64-minimal.iso** aus.
Ganz unten wird der vorher formatierte USB-Stick ausgewählt.
Wichtig: Das Häkchen bei "Install from RS-232 Serial Console" aktivieren und die Baudrate auf 115200 erhöhen.



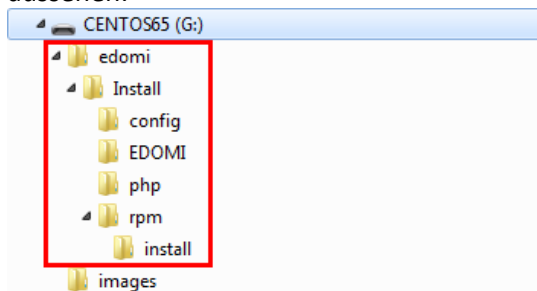
3. Anschliessend auf OK klicken und der Stick wird in mehreren Schritten vorbereitet. Dies kann einige Minuten dauern.



4. Am Ende kommt die Bestätigung, dass der Stick nun vorbereitet ist. Wir könnten somit bereits damit booten und CentOS installieren. Doch da wir den Stick gerade noch am Computer angeschlossen haben, macht es Sinn, das EDOMI-Installationspaket darauf zu kopieren. Das Tool vorher mit **Exit** beenden.



5. Das aktuelle EDOMI Installationspaket von www.edomi.de downloaden und das ZIP entpacken. Den kompletten Ordner "edomi" auf den USB-Stick (Root-Verzeichnis) kopieren. Es sollte dann in etwa so aussehen:

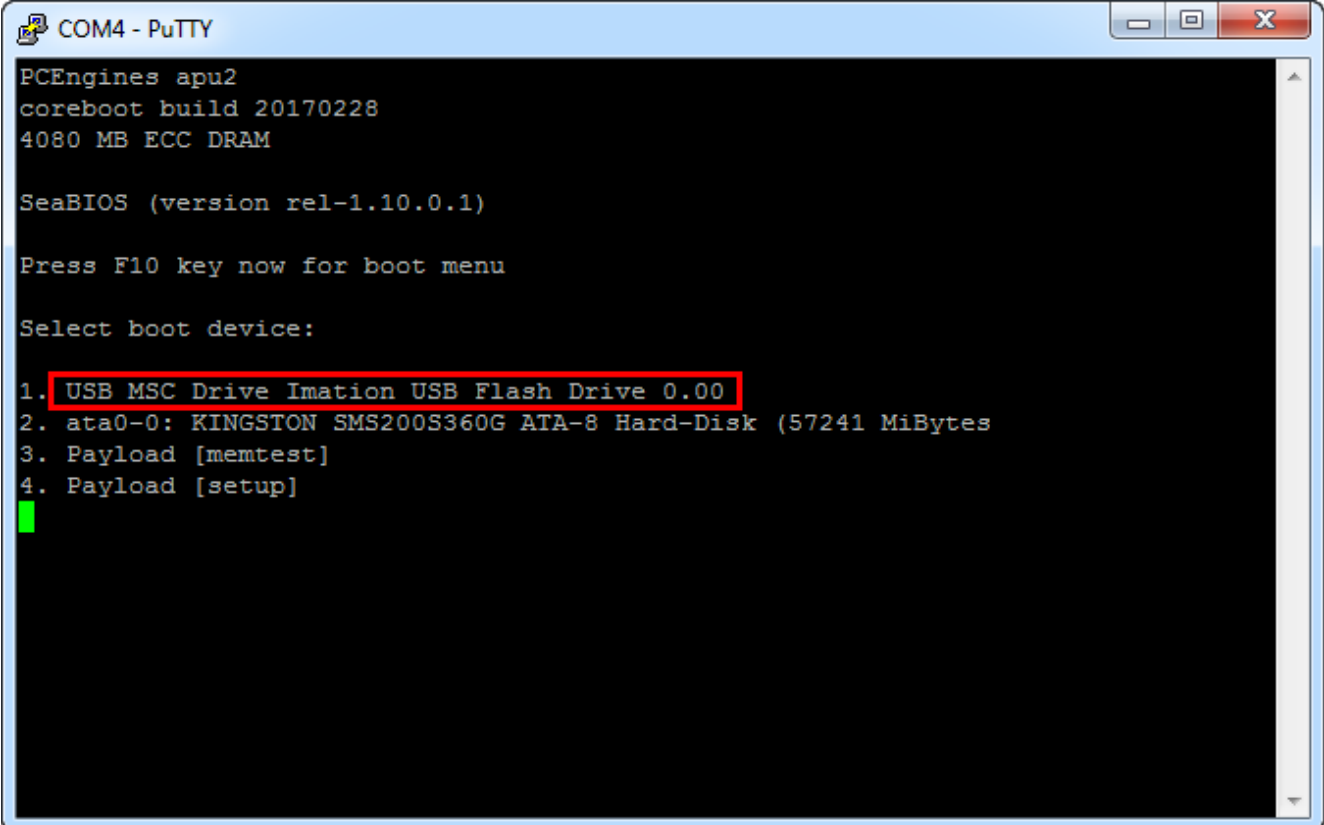


Name	Änderungsdatum	Typ	Größe
edomi	28.04.2017 16:43	Dateiordner	
images	28.04.2017 16:35	Dateiordner	
boot	29.11.2013 13:14	Sicherheitskatalog	2 KB
boot	29.11.2013 13:04	Outlook Item	1 KB
grub.conf	27.11.2013 22:52	CONF-Datei	1 KB
initrd	27.11.2013 22:52	IrfanView IMG File	32'611 KB
linux	28.04.2017 16:35	Virtual CloneDrive	407'552 KB
memtest	29.11.2013 13:04	Datei	162 KB
splash	29.11.2013 13:04	IrfanView JPG File	148 KB
syslinux.cfg	28.04.2017 16:40	CFG-Datei	1 KB
TRANS.TBL	29.11.2013 13:14	TBL-Datei	3 KB
vesamenu.c32	28.04.2017 16:40	C32-Datei	145 KB
vmlinuz	29.11.2013 13:04	Datei	4'032 KB

CentOS 6.5 minimal installieren

Da wir nun den bootfähigen USB-Stick erstellt haben, können wir mit der Installation beginnen. Auch hier ist es wichtig, dass wir VORHER eine funktionierende Putty-Verbindung aufbauen.

1. Bei aktiver Putty-Verbindung das apu2-System am 220V ausschalten, den vorbereiteten USB-Stick am apu2-System einstecken und das Netzteil wieder mit Strom versorgen.
2. Nach ein paar Sekunden sollte das apu2 **Bootmenü** erscheinen. Abhängig von den installierten Komponenten kann das Bild etwas abweichen. In meinem Fall ist 1 der USB-Stick und 2 die interne SSD. Da wir vom USB-Stick booten müssen, wählen wir die entsprechende Zahl aus (hier 1).



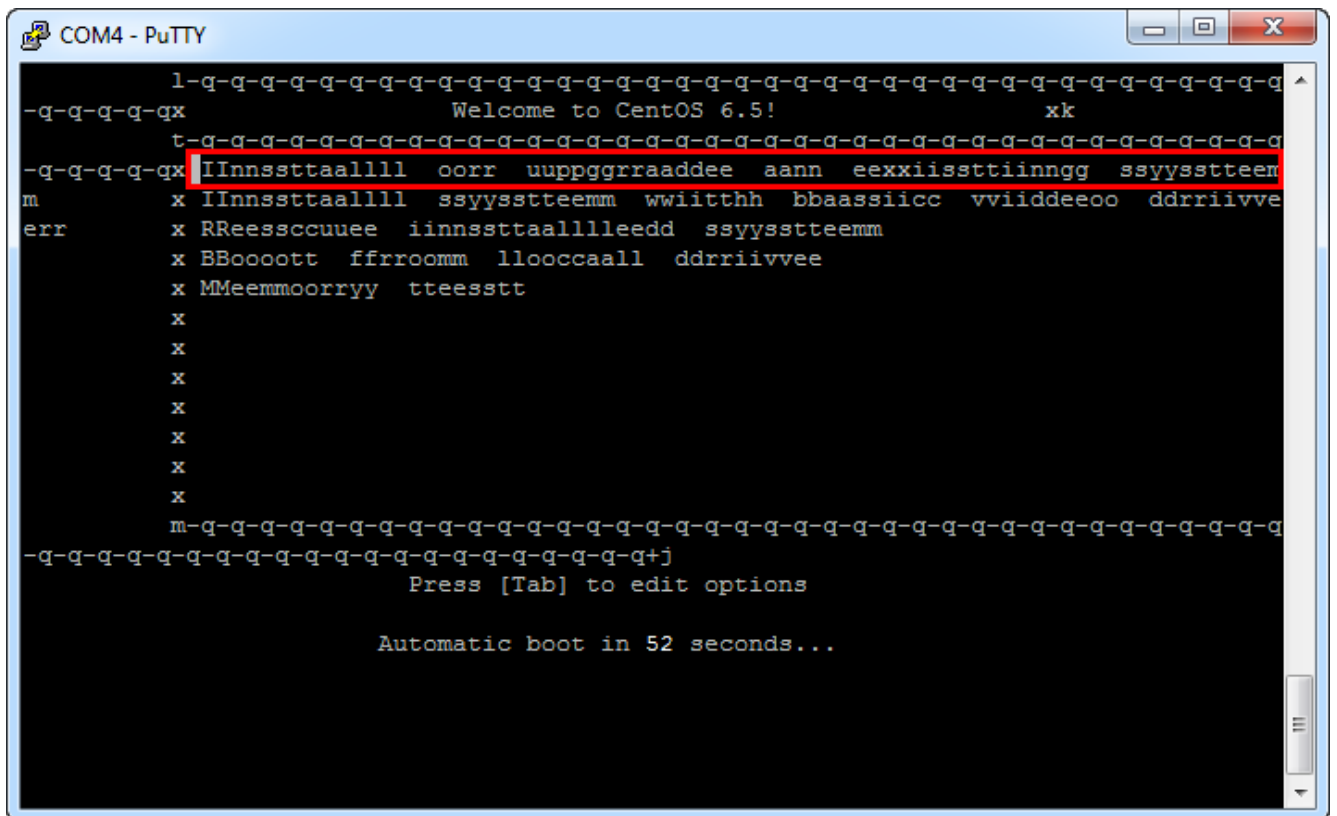
```
COM4 - PuTTY
PCEngines apu2
coreboot build 20170228
4080 MB ECC DRAM

SeaBIOS (version rel-1.10.0.1)

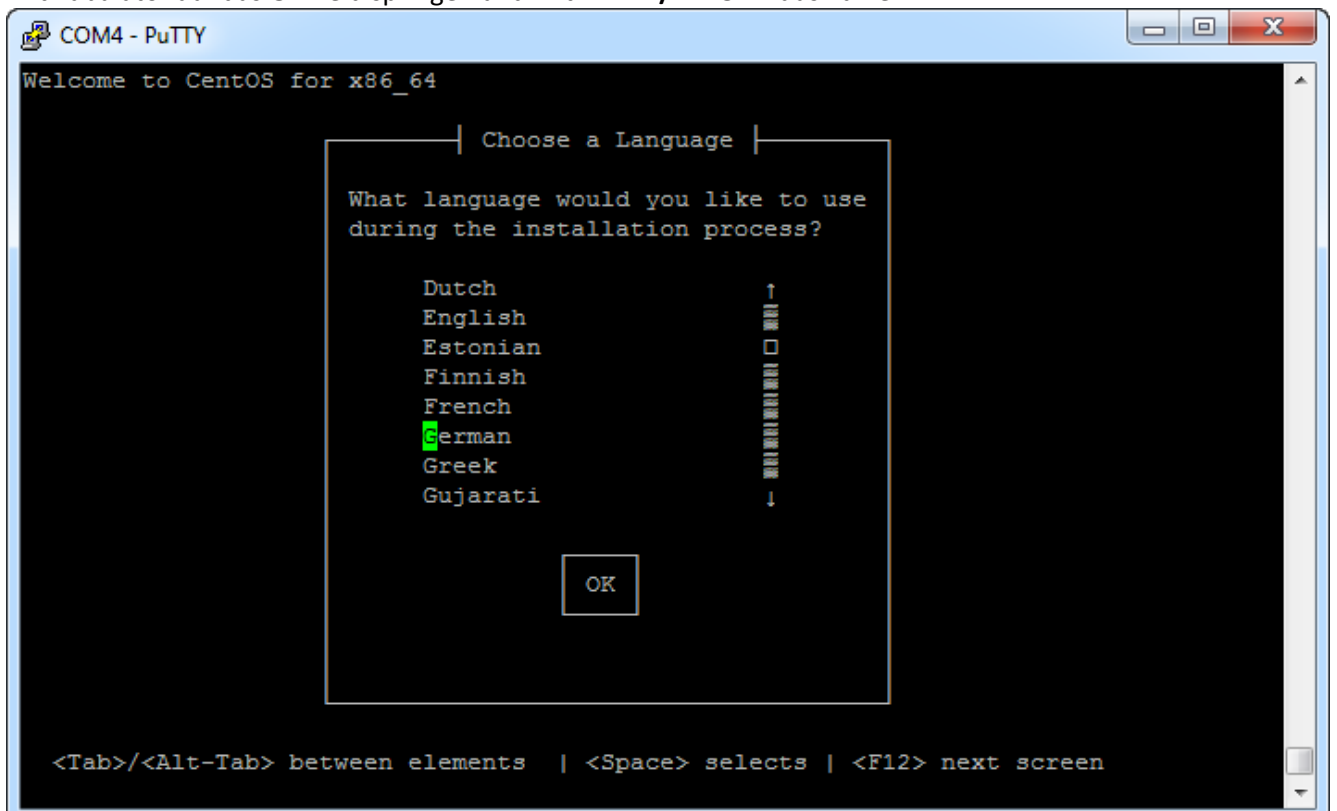
Press F10 key now for boot menu

Select boot device:
1. USB MSC Drive Imation USB Flash Drive 0.00
2. ata0-0: KINGSTON SMS200S360G ATA-8 Hard-Disk (57241 MiBytes)
3. Payload [memtest]
4. Payload [setup]
█
```

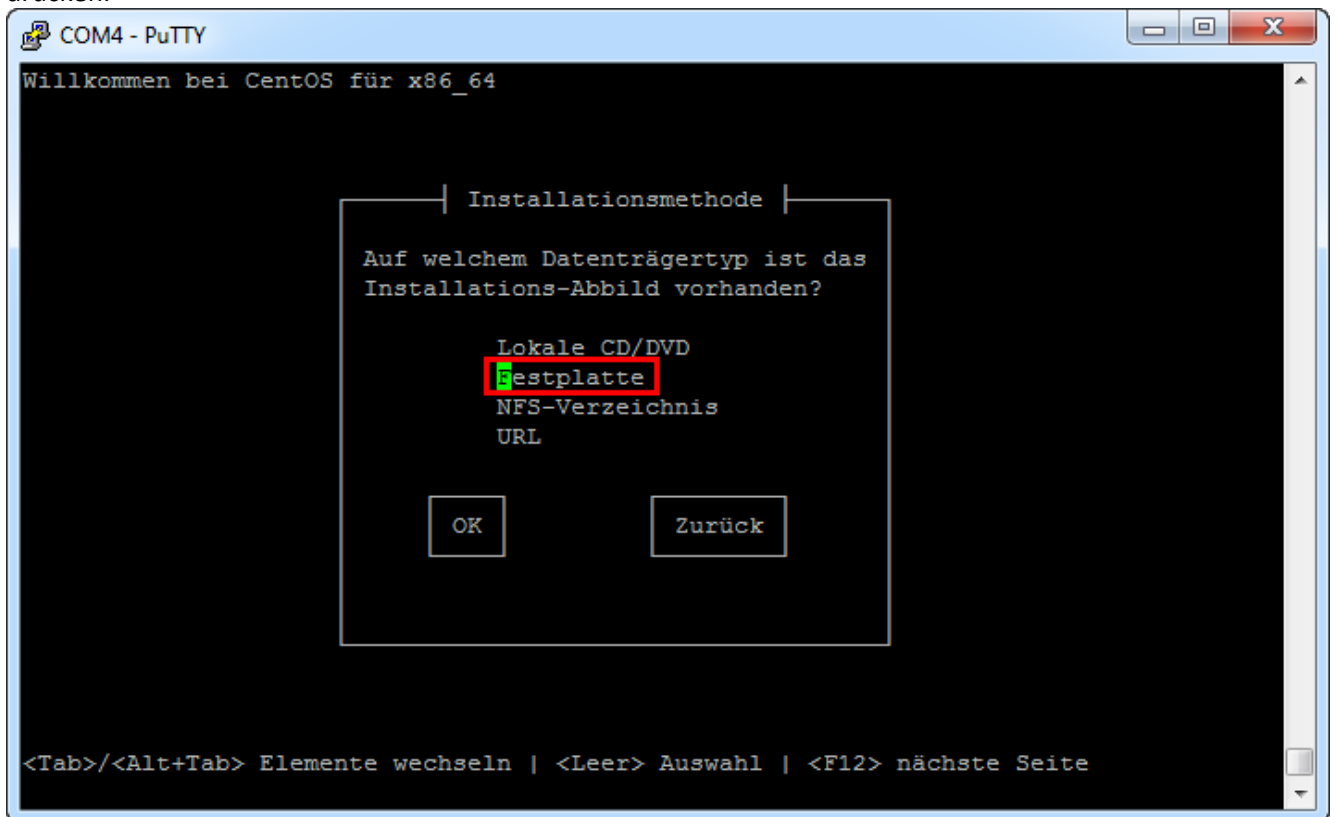
3. Nun erscheint das Installationsmenü von CentOS. Das erste Bild wird komischerweise mit falschen Echowerten dargestellt, d.h. alle Buchstaben werden doppelt oder mehrfach angezeigt. Trotz dieser etwas verwirrenden Ansicht kann man die Optionen durchaus lesen:
 - Install or upgrade an existing system
 - Install system with basic video driver
 - Rescue installed system
 - Boot from local drive
 - Memory testIn unserem Fall wählen wir mit dem Cursor die **erste Zeile** aus (auf den Cursorbalken achten) und drücken die **ENTER/RETURN**-Taste.



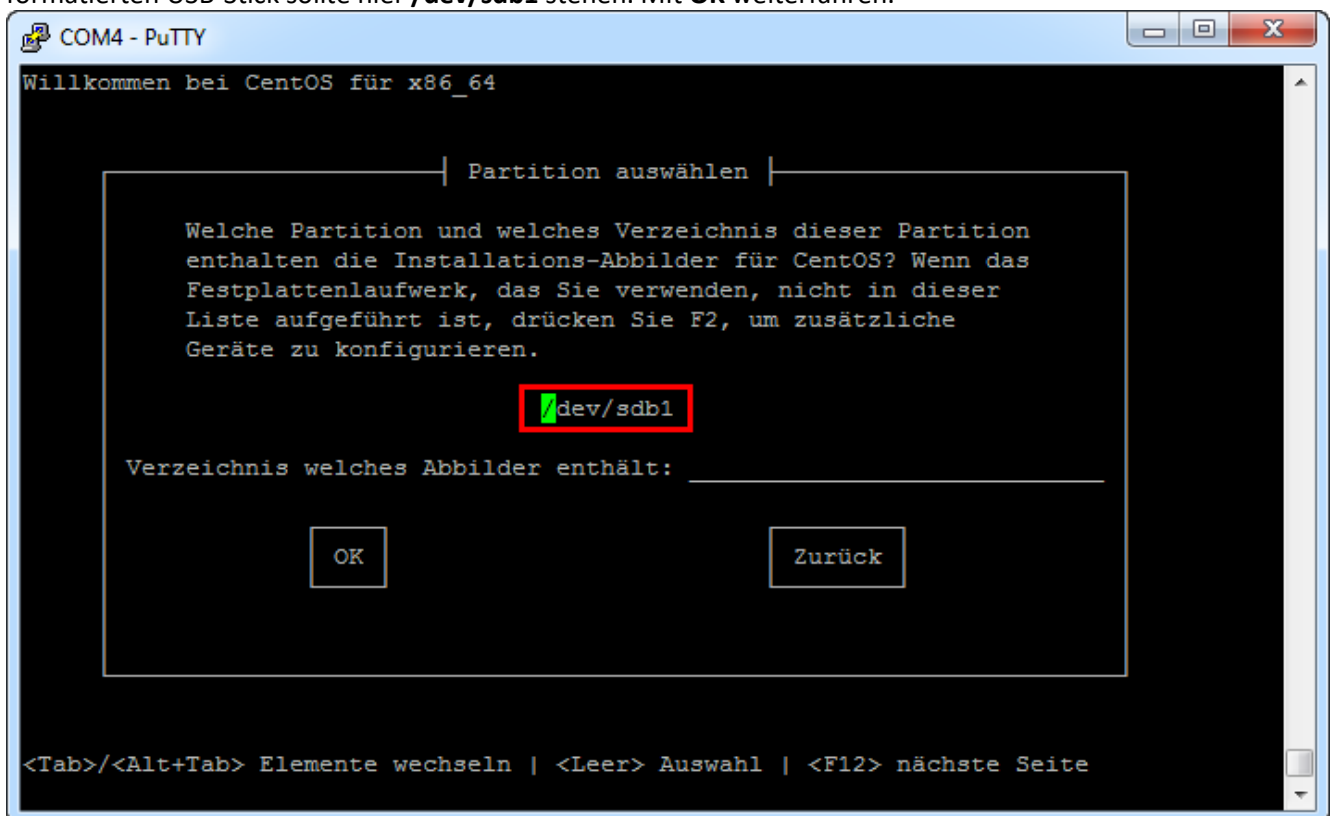
4. Nun sieht es schon wieder etwas besser aus (glücklicherweise bleibt es ab jetzt auch dabei). Mit dem Cursor (grüner Cursorbalken) können wir die gewünschte Sprache auswählen. Für diese Anleitung habe ich German gewählt (in meinem produktiven System verwende ich jedoch englisch). Mit der Tabulator-Taste kann man zwischen den einzelnen Elementen hin- und herspringen. Sollte der grüne Cursor in einer Auswahl-Liste nicht mehr angezeigt werden, so gilt immer die Option, die vor dem Drücken von Tabulator ausgewählt war! Mit Tabulator auf das **OK**-Feld springen und mit **ENTER/RETURN** auswählen.



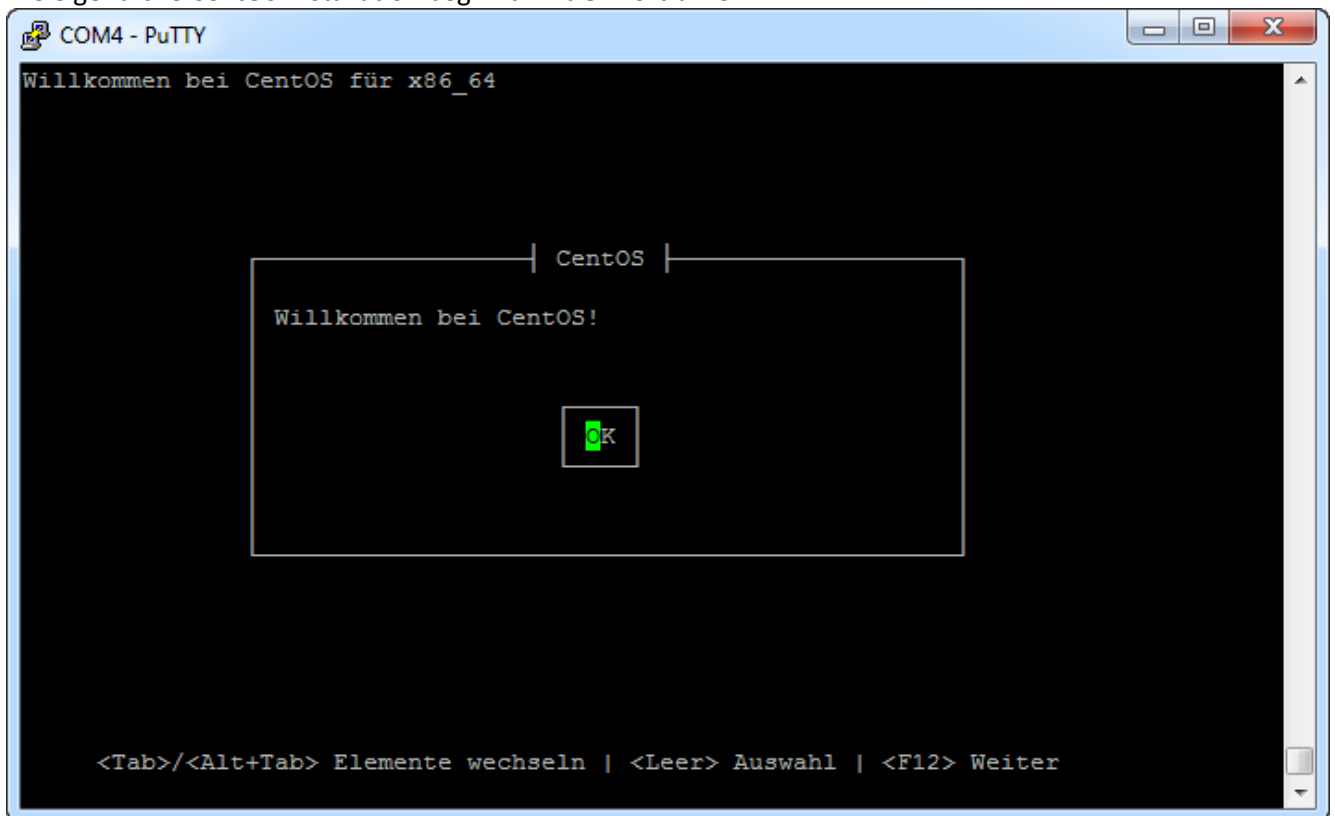
5. Hier wählen wir den Datenträgertyp aus, auf dem sich die CentOS-Installationsdateien befinden. Da der USB-Stick in diesem Fall als "Festplatte" definiert ist, müssen wir **Festplatte** auswählen und danach auf **OK** drücken.



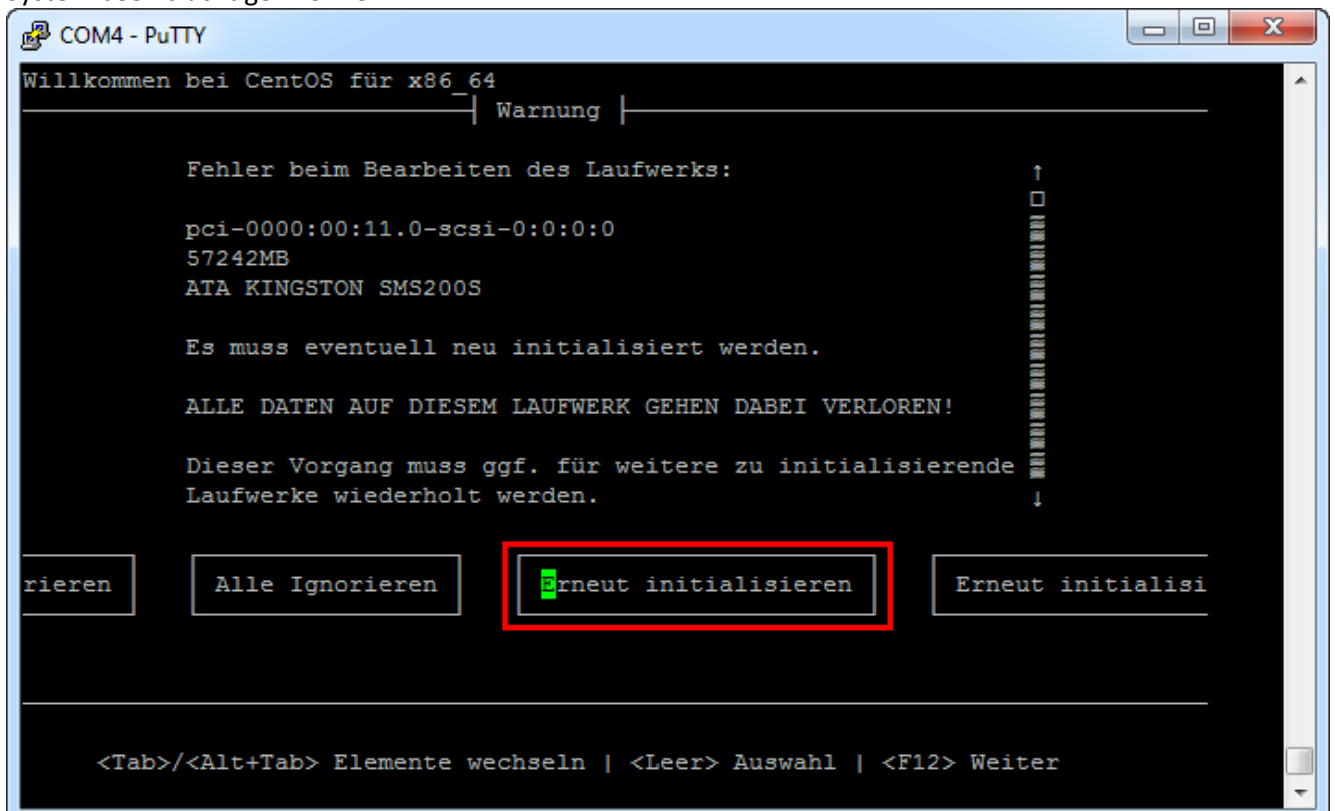
6. Hier muss die Partition der CentOS-Installationsdateien ausgewählt werden. Bei einem normal formatierten USB-Stick sollte hier **/dev/sdb1** stehen. Mit **OK** weiterfahren.



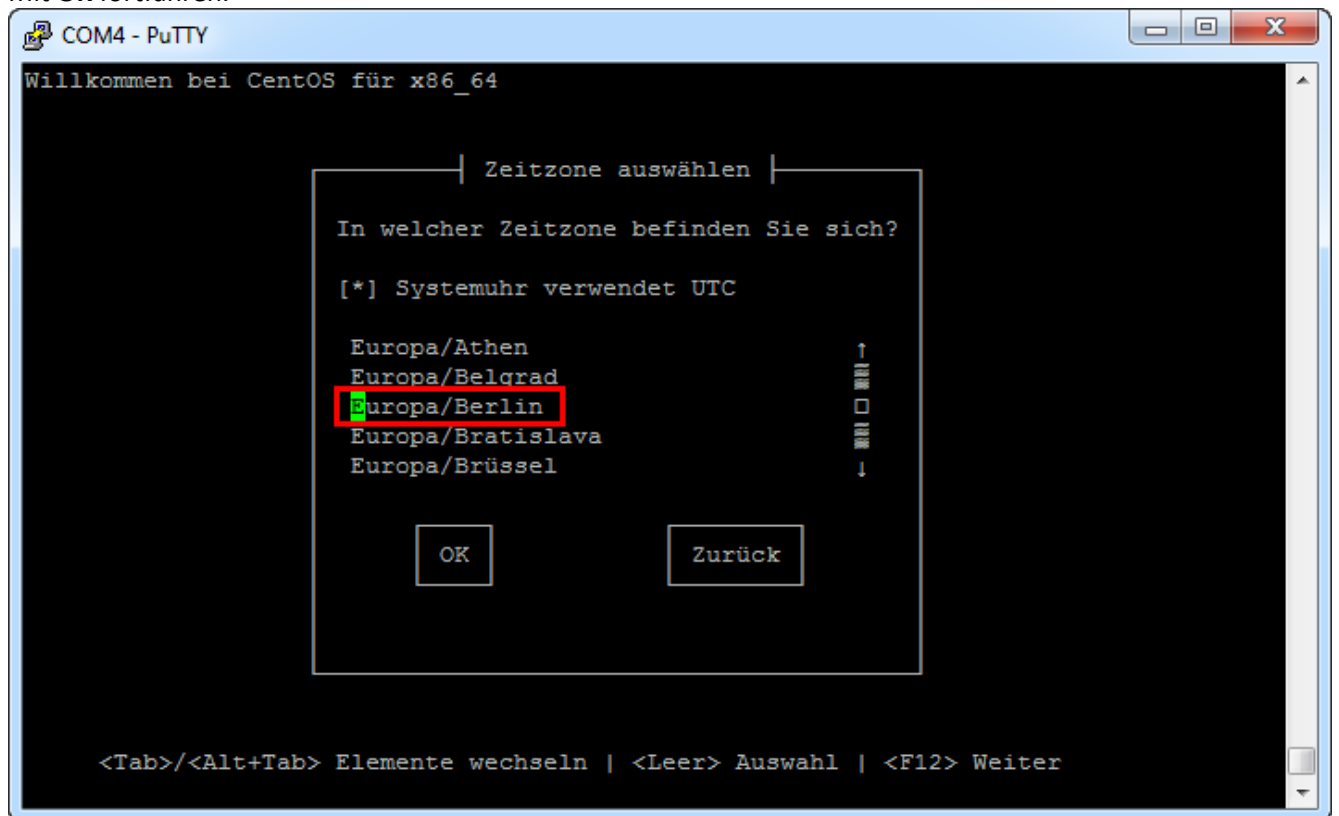
7. Die eigentliche CentOS-Installation beginnt. Mit **OK** fortfahren.



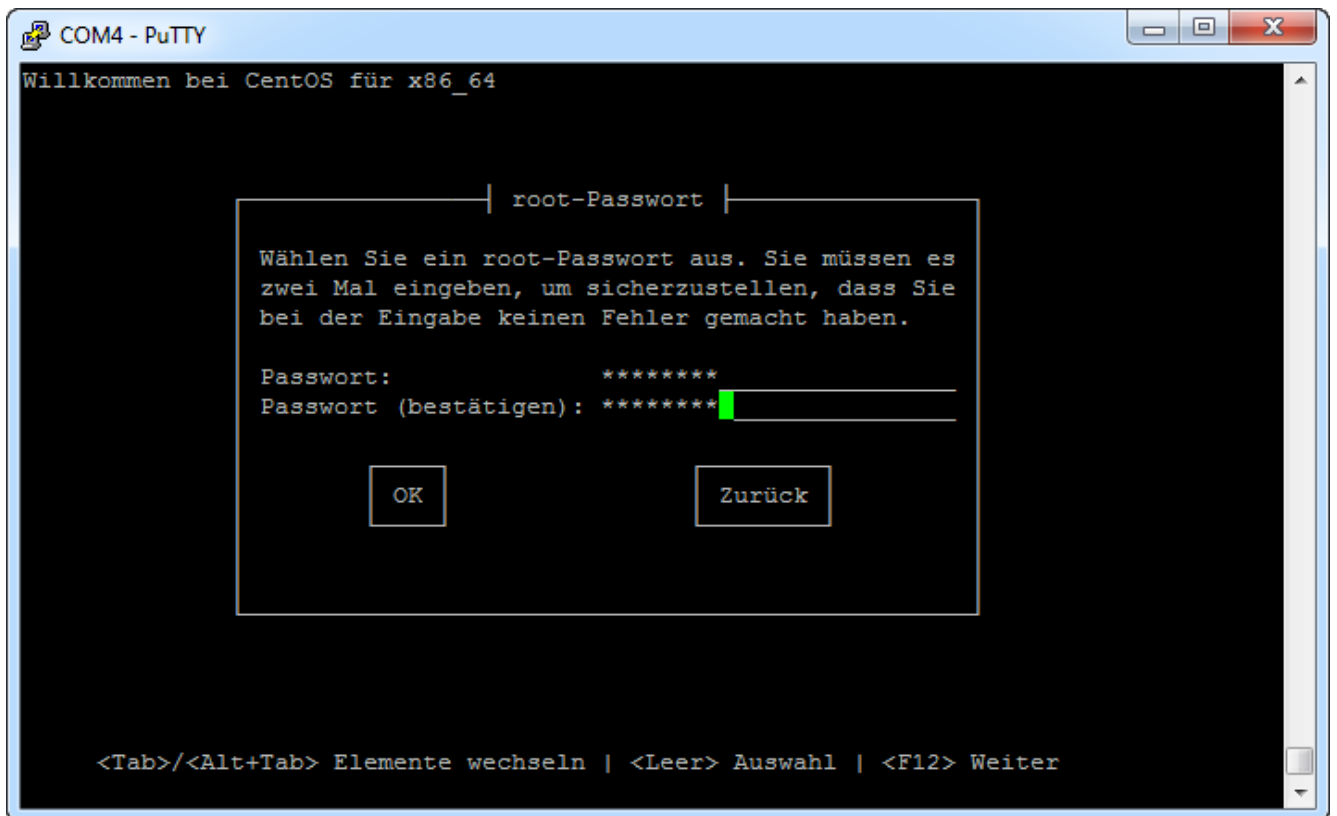
8. Als Ziel wollen wir CentOS auf die interne Disk (hier eine 60 GB SSD) installieren. Abhängig, ob die Disk bereits formatiert wurde, können hier evtl. andere Anzeigen erscheinen. In meinem Fall war die Disk leer und deshalb möchte ich sie mit "**Erneut initialisieren**" neu formatieren. (Dabei gehen natürlich alle Daten auf der SSD verloren!)
Wenn möglich, sollte man immer die Disk vorher löschen, damit nicht fehlerhafte Altlasten das neue System beeinträchtigen können.



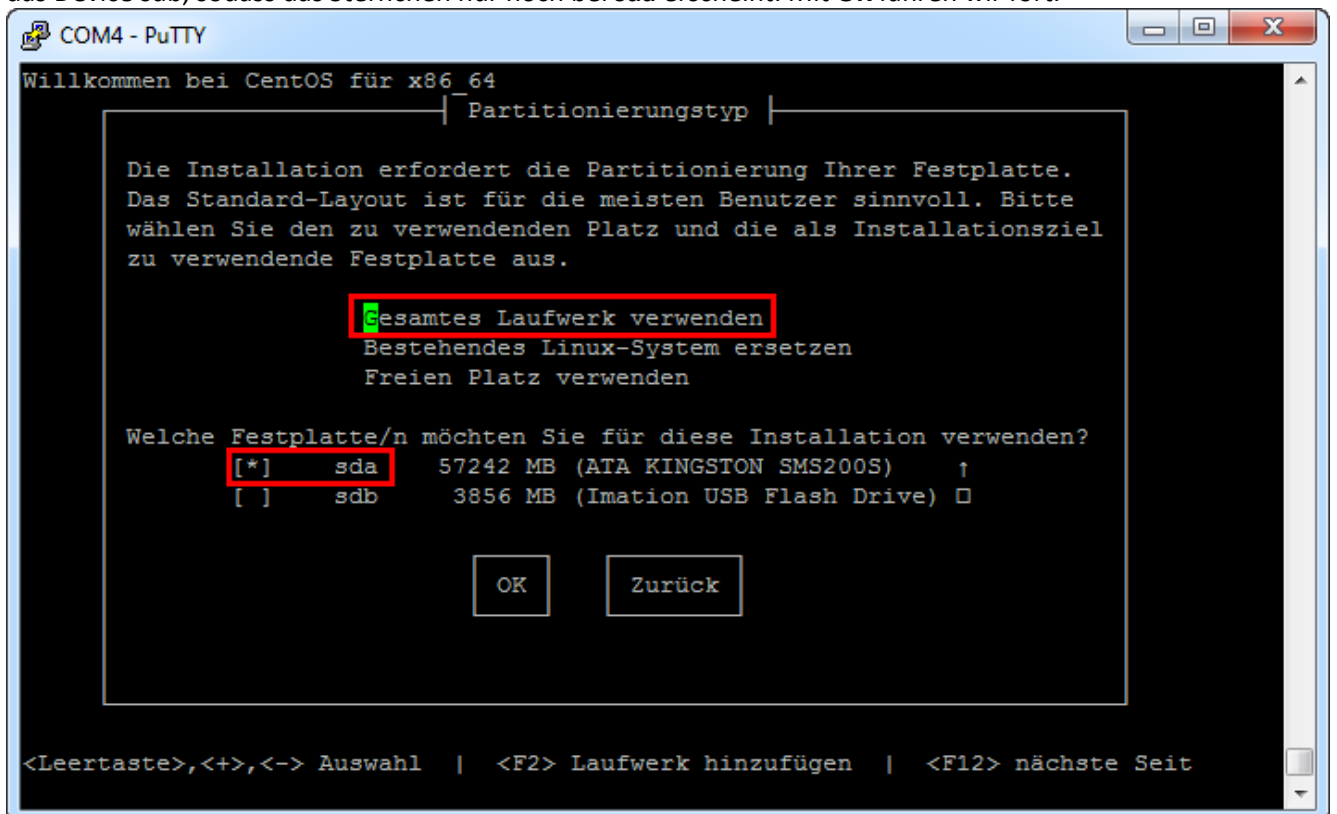
9. Nun kann die Zeitzone ausgewählt werden. Kurz die Taste "e" drücken und dann runterscrollen, bis "Europa/Berlin" erscheint (oder in welcher Zeitzone man sich halt befindet). In EDOMI sollte man später die gleiche Zeitzone verwenden! Mit **OK** fortfahren.



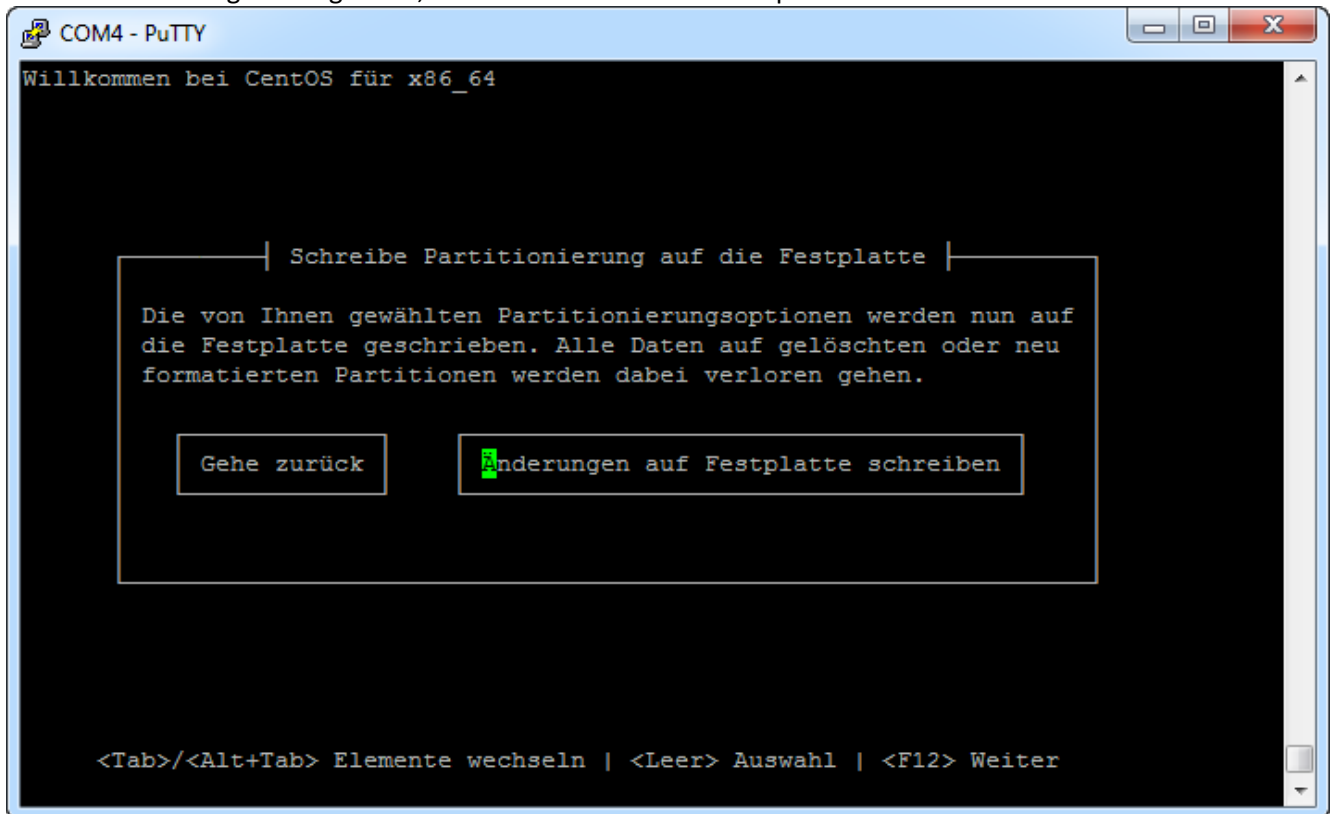
10. Jetzt müssen wir ein root-Passwort für das CentOS-Betriebssystem erzeugen. Auch wenn man das EDOMI-System nie direkt am Internet betreiben sollte, ist es dennoch sinnvoll, ein starkes Passwort einzugeben. Der root-Account und dieses Passwort wird immer benötigt, wenn man direkt auf dem CentOS-Betriebssystem zu tun hat (z.B. auch Zugriff via SSH etc.). **Achtung: Der root-Account hat die höchsten Privilegien!** Mit **OK** fortfahren.



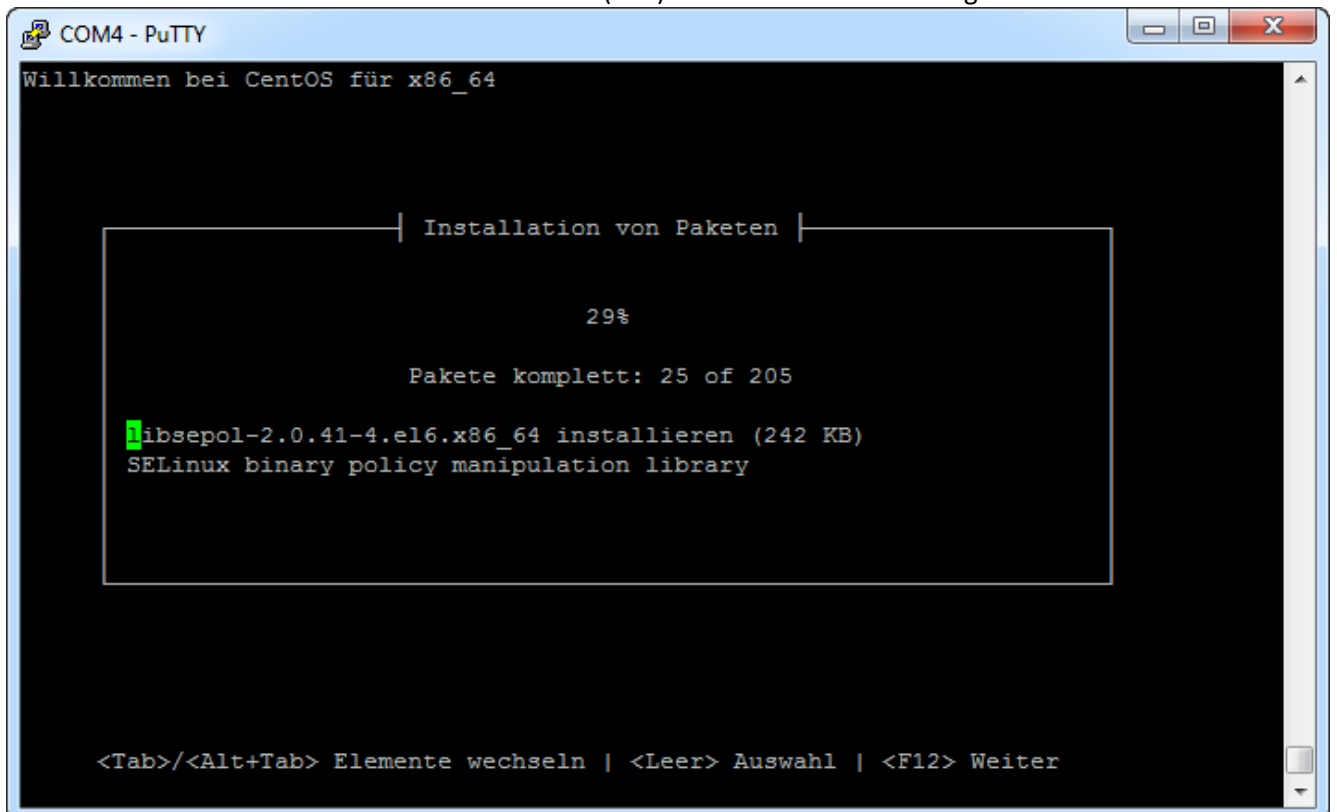
11. Hier können wir wählen, ob beim Zielmedium (interne SSD) das gesamte Laufwerk verwendet werden soll. Dies ist natürlich in unserem Sinn, deshalb wählen wir mit dem grünen Cursor "**Gesamtes Laufwerk verwenden**" aus. Mit Tabulator wechseln wir in den unteren Bereich und deaktivieren sicherheitshalber das Device sdb, sodass das Sternchen nur noch bei **sda** erscheint. Mit **OK** fahren wir fort.



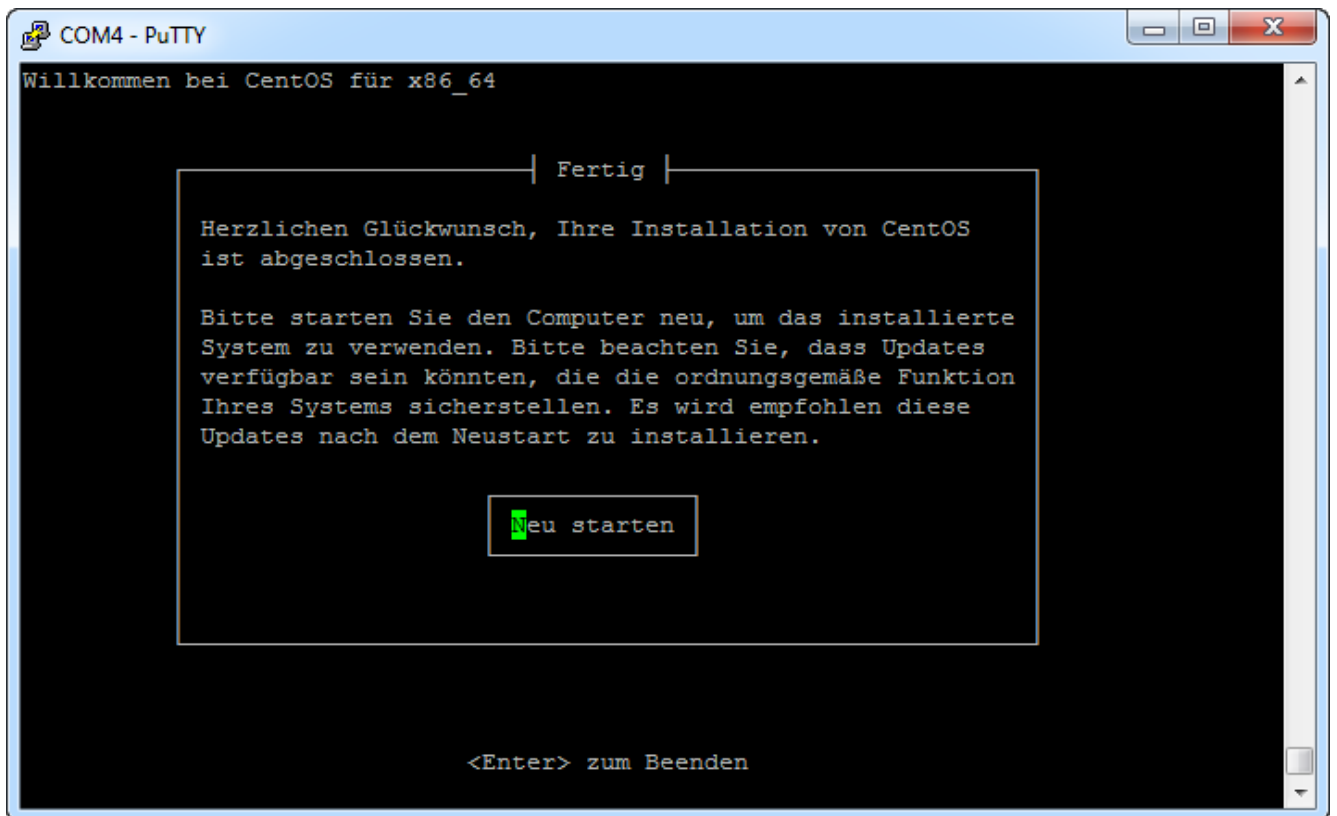
12. Mit dieser Meldung bestätigen wir, dass wir die interne SSD komplett löschen werden.



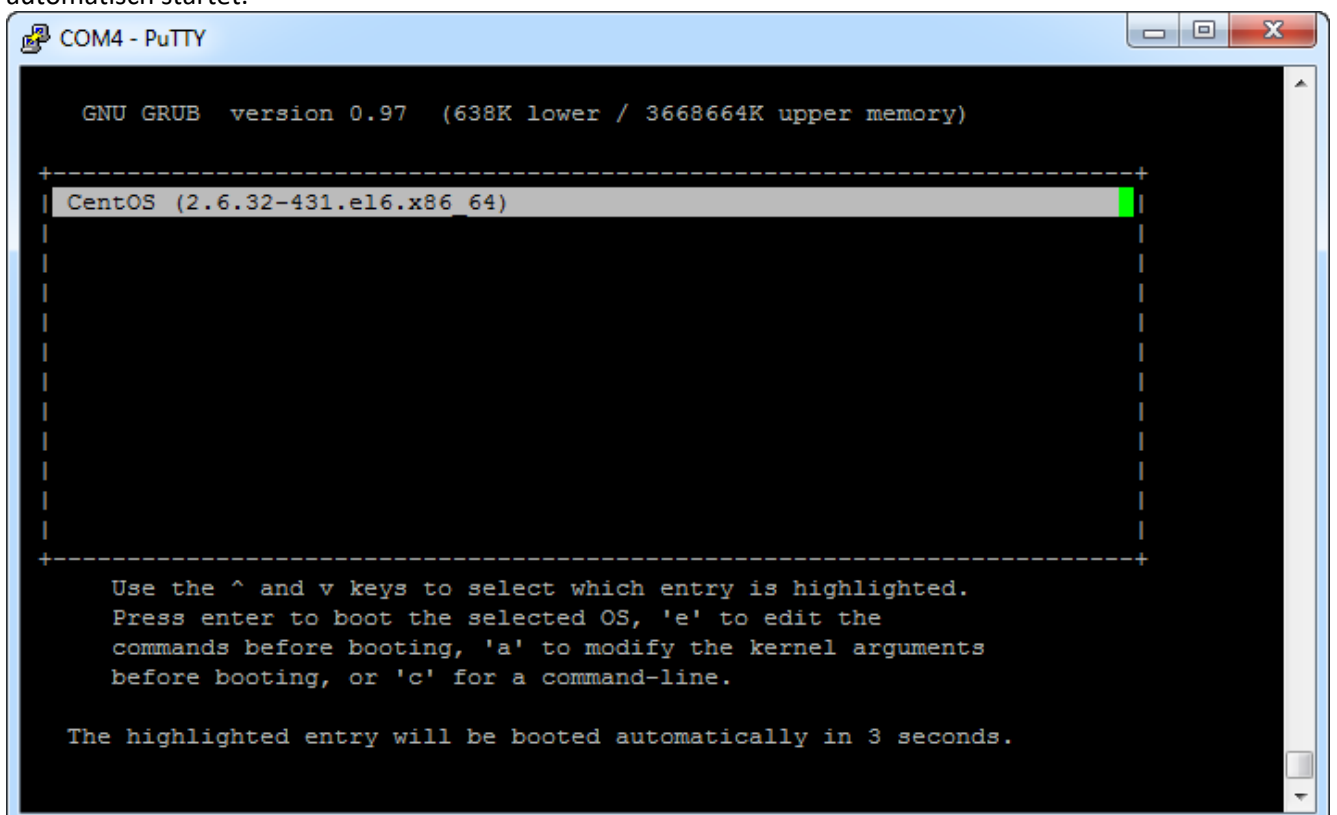
13. Nun wird CentOS 6.5 minimal auf die interne Disk (SSD) installiert. Dies kann einige Minuten dauern.



14. Erscheint diese Meldung, so ist die Installation abgeschlossen. Mit **Neu starten** können wir das apu2-System rebooten. (Entgegen der angezeigten Empfehlung sollte keine CentOS-Updates installiert werden!)
Es empfiehlt sich, den USB-Stick sofort nach dem Neustart auszustechen oder im Bootmenü dann die interne Disk (vermutlich Punkt 2) auszuwählen.



15. Nach dem Neustart (von der internen Disk) erscheint nach ein paar Sekunden ein weiteres Bootmenü. Da wir nur eine Option haben, können wir entweder CentOS auswählen oder warten, bis der Countdown es automatisch startet.



16. Nach der Bootphase mit vielen Linux-Meldungen sollte irgendwann der **Login-Prompt** erscheinen. Wir haben es geschafft und könnten uns hier mit dem Account root und dem früher definierten Passwort am CentOS anmelden. Die Installation von CentOS 6.5 minimal ist damit abgeschlossen.

```
COM4 - PuTTY
Betrete nicht-interaktiven Start-Modus
Starting monitoring for VG VolGroup: 3 logical volume(s) in volume group "VolGroup"
monitored
[ OK ]
ip6tables: Firewall-Regeln anwenden: NET: Registered protocol family 10
lo: Disabled Privacy Extensions
ip6_tables: (C) 2000-2006 Netfilter Core Team
nf_contrack version 0.5.0 (16384 buckets, 65536 max)
[ OK ]
iptables: Firewall-Regeln anwenden: ip_tables: (C) 2000-2006 Netfilter Core Team
[ OK ]
Loopback-Schnittstelle hochfahren: [ OK ]
auditd starten: [ OK ]
System-Protokollierdienst starten: [ OK ]
Mounting filesystems: [ OK ]
Erneutes Triggering für fehlgeschlagene udev-Ereignisse[ OK ]
sshd starten: [ OK ]
Postfix starten: [ OK ]
crond starten: [ OK ]

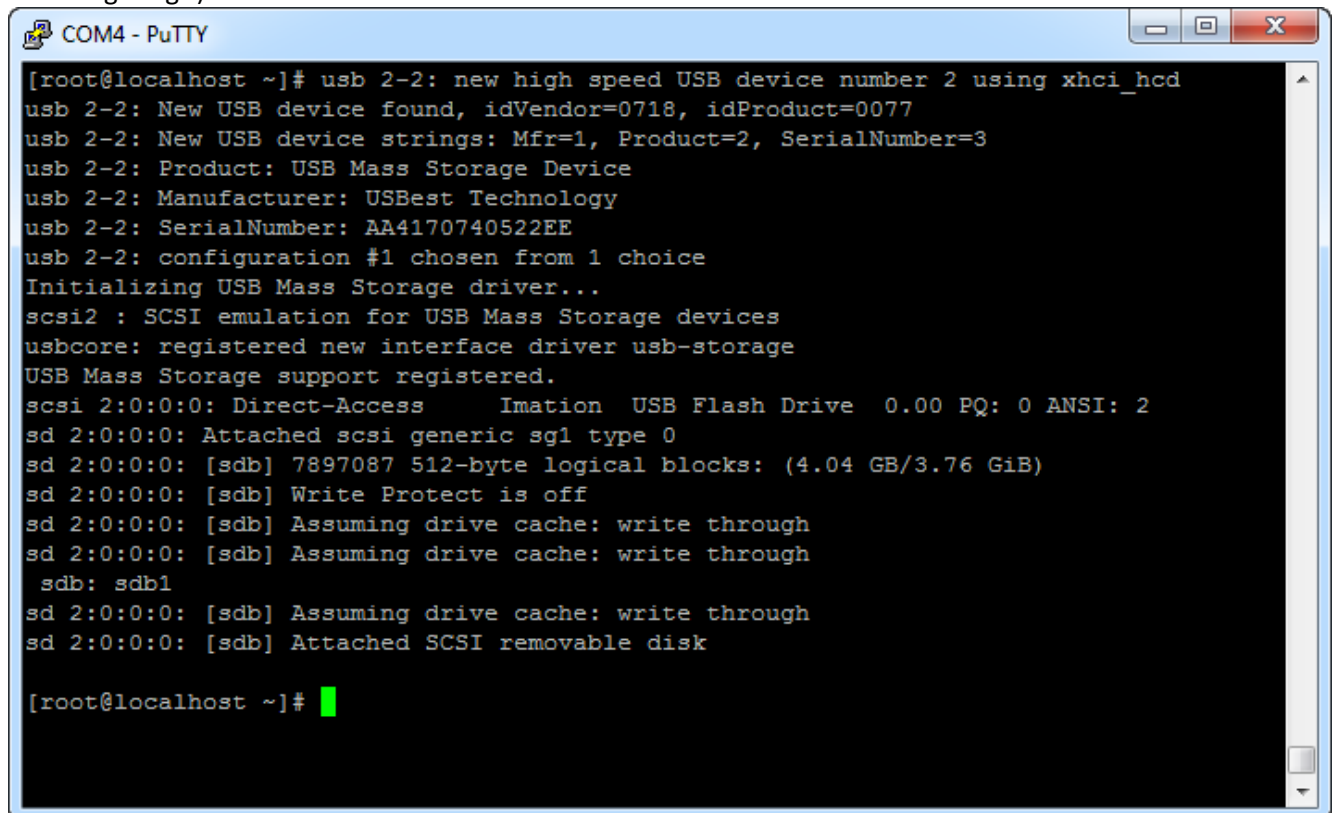
CentOS release 6.5 (Final)
Kernel 2.6.32-431.el6.x86_64 on an x86_64

localhost.localdomain login: █
```

EDOMI installieren

Da wir nun den bootfähigen USB-Stick erstellt haben, können wir mit der Installation beginnen. Auch hier ist es wichtig, dass wir VORHER eine funktionierende Putty-Verbindung aufbauen.

1. CentOS starten und sich mit dem **root** Account anmelden.
2. Wird der USB-Stick mit der EDOMI-Installation erst jetzt eingesteckt, so wird dies sofort in der Konsole mit einigen Textzeilen angezeigt. (Wurde der Stick bereits beim Starten eingesteckt, so wird dies hier natürlich nicht angezeigt.)



```
COM4 - PuTTY
[root@localhost ~]# usb 2-2: new high speed USB device number 2 using xhci_hcd
usb 2-2: New USB device found, idVendor=0718, idProduct=0077
usb 2-2: New USB device strings: Mfr=1, Product=2, SerialNumber=3
usb 2-2: Product: USB Mass Storage Device
usb 2-2: Manufacturer: USBest Technology
usb 2-2: SerialNumber: AA4170740522EE
usb 2-2: configuration #1 chosen from 1 choice
Initializing USB Mass Storage driver...
scsi2 : SCSI emulation for USB Mass Storage devices
usbcore: registered new interface driver usb-storage
USB Mass Storage support registered.
scsi 2:0:0:0: Direct-Access          Imation USB Flash Drive  0.00 PQ: 0 ANSI: 2
sd 2:0:0:0: Attached scsi generic sql type 0
sd 2:0:0:0: [sdb] 7897087 512-byte logical blocks: (4.04 GB/3.76 GiB)
sd 2:0:0:0: [sdb] Write Protect is off
sd 2:0:0:0: [sdb] Assuming drive cache: write through
sd 2:0:0:0: [sdb] Assuming drive cache: write through
sdb: sdb1
sd 2:0:0:0: [sdb] Assuming drive cache: write through
sd 2:0:0:0: [sdb] Attached SCSI removable disk

[root@localhost ~]# █
```

3. Bevor wir EDOMI installieren können, müssen wir wieder den USB-Stick mounten. Auch wenn wir den Device-Namen unterdessen kennen dürften, so gibt uns der Befehl

fdisk -l Sicherheit. In unserem Fall wäre dies also wieder **/dev/sdb1**.

```
COM4 - PuTTY
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

Platte /dev/mapper/VolGroup-lv_home: 4924 MByte, 4924112896 Byte
255 Köpfe, 63 Sektoren/Spur, 598 Zylinder
Einheiten = Zylinder von 16065 × 512 = 8225280 Bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

Platte /dev/sdb: 4043 MByte, 4043308544 Byte
255 Köpfe, 63 Sektoren/Spur, 491 Zylinder
Einheiten = Zylinder von 16065 × 512 = 8225280 Bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

  Gerät boot.   Anfang      Ende      Blöcke   Id System
-----
/dev/sdb1 *      1          492     3948416  b  W95 FAT32
Partition 1 hat unterschiedliche phys./log. Enden:
  phys=(490, 254, 63) logisch=(491, 143, 36)
[root@localhost ~]#
```

4. Im Unterschied zur (optionalen) Variante mit dem BIOS-Update USB-Stick müssen wir hier zuerst einen sogenannten Mountpoint erstellen. Dies ist nur ein Ordner, den wir am besten im Ordner **/media** erstellen. Der Befehl dazu lautet:

mkdir /media/usb

```
COM4 - PuTTY
Disk identifier: 0x00000000

Platte /dev/mapper/VolGroup-lv_home: 4924 MByte, 4924112896 Byte
255 Köpfe, 63 Sektoren/Spur, 598 Zylinder
Einheiten = Zylinder von 16065 × 512 = 8225280 Bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

Platte /dev/sdb: 4043 MByte, 4043308544 Byte
255 Köpfe, 63 Sektoren/Spur, 491 Zylinder
Einheiten = Zylinder von 16065 × 512 = 8225280 Bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

  Gerät boot.   Anfang      Ende      Blöcke   Id System
-----
/dev/sdb1 *      1          492     3948416  b  W95 FAT32
Partition 1 hat unterschiedliche phys./log. Enden:
  phys=(490, 254, 63) logisch=(491, 143, 36)
[root@localhost ~]# mkdir /media/usb
[root@localhost ~]#
```

5. Anschliessend folgt wieder der bekannte mount-Befehl. Der Device-Name des USB-Sticks kennen wir ja bereits und der Mountpoint haben wir soeben angelegt.

mount /dev/sdb1 /media/usb


```
COM4 - PuTTY

Platte /dev/mapper/VolGroup-lv_home: 4924 MByte, 4924112896 Byte
255 Köpfe, 63 Sektoren/Spur, 598 Zylinder
Einheiten = Zylinder von 16065 × 512 = 8225280 Bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

Platte /dev/sdb: 4043 MByte, 4043308544 Byte
255 Köpfe, 63 Sektoren/Spur, 491 Zylinder
Einheiten = Zylinder von 16065 × 512 = 8225280 Bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

  Gerät boot.      Anfang      Ende      Blöcke    Id System
/dev/sdb1 *          1          492     3948416   b  W95 FAT32
Partition 1 hat unterschiedliche phys./log. Enden:
  phys=(490, 254, 63) logisch=(491, 143, 36)
[root@localhost ~]# mkdir /media/usb
[root@localhost ~]# mount /dev/sdb1 /media/usb
[root@localhost ~]#
```

6. Zur Sicherheit zeigen wir uns kurz den Inhalt des nun gemounteten USB-Sticks an, damit wir sicher sind, dass wir auf die EDOMI-Installationsdateien zugreifen können. Mit dem Befehl `ls -l /media/usb` sehen wir hoffentlich den EDOMI-Ordner.

```
COM4 - PuTTY

  Gerät boot.      Anfang      Ende      Blöcke    Id System
/dev/sdb1 *          1          492     3948416   b  W95 FAT32
Partition 1 hat unterschiedliche phys./log. Enden:
  phys=(490, 254, 63) logisch=(491, 143, 36)
[root@localhost ~]# mkdir /media/usb
[root@localhost ~]# mount /dev/sdb1 /media/usb
[root@localhost ~]# ls -l /media/usb
insgesamt 444700
-rwxr-xr-x. 1 root root      2048 29. Nov 2013 boot.cat
-rwxr-xr-x. 1 root root         84 29. Nov 2013 boot.msg
drwxr-xr-x. 3 root root     4096 28. Apr 16:43 edomi
-rwxr-xr-x. 1 root root      334 27. Nov 2013 grub.conf
drwxr-xr-x. 2 root root     4096 28. Apr 16:35 images
-rwxr-xr-x. 1 root root 33392971 27. Nov 2013 initrd.img
-r-xr-xr-x. 1 root root    13639 28. Apr 16:40 ldlinux.sys
-rwxr-xr-x. 1 root root 417333248 28. Apr 16:35 linux.iso
-rwxr-xr-x. 1 root root    165080 29. Nov 2013 memtest
-rwxr-xr-x. 1 root root    151230 29. Nov 2013 splash.jpg
-rwxr-xr-x. 1 root root     1015 28. Apr 16:40 syslinux.cfg
-rwxr-xr-x. 1 root root     2215 29. Nov 2013 TRANS.TBL
-rwxr-xr-x. 1 root root    147996 28. Apr 16:40 vesamenu.c32
-rwxr-xr-x. 1 root root   4128368 29. Nov 2013 vmlinuz
[root@localhost ~]#
```

7. Mit dem Befehl `cd /media/usb/edomi/Install` wechseln wir direkt in das entsprechende EDOMI-Installationsverzeichnis. (Achtung:

Gross/Kleinschreibung beachten! Das I von Install ist gross.)

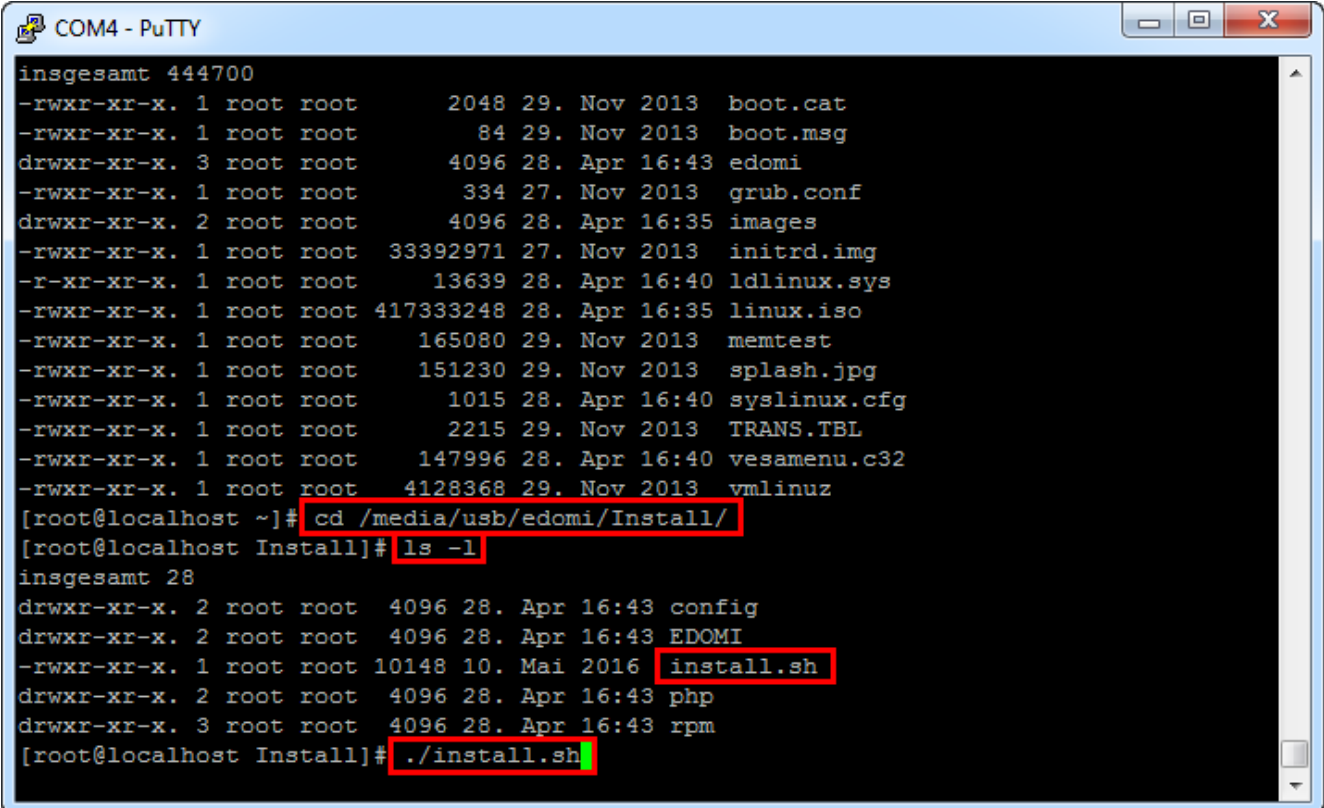
Anschliessend können wir mit

```
ls -l
```

nochmals den Inhalt anzeigen und vorallem mit

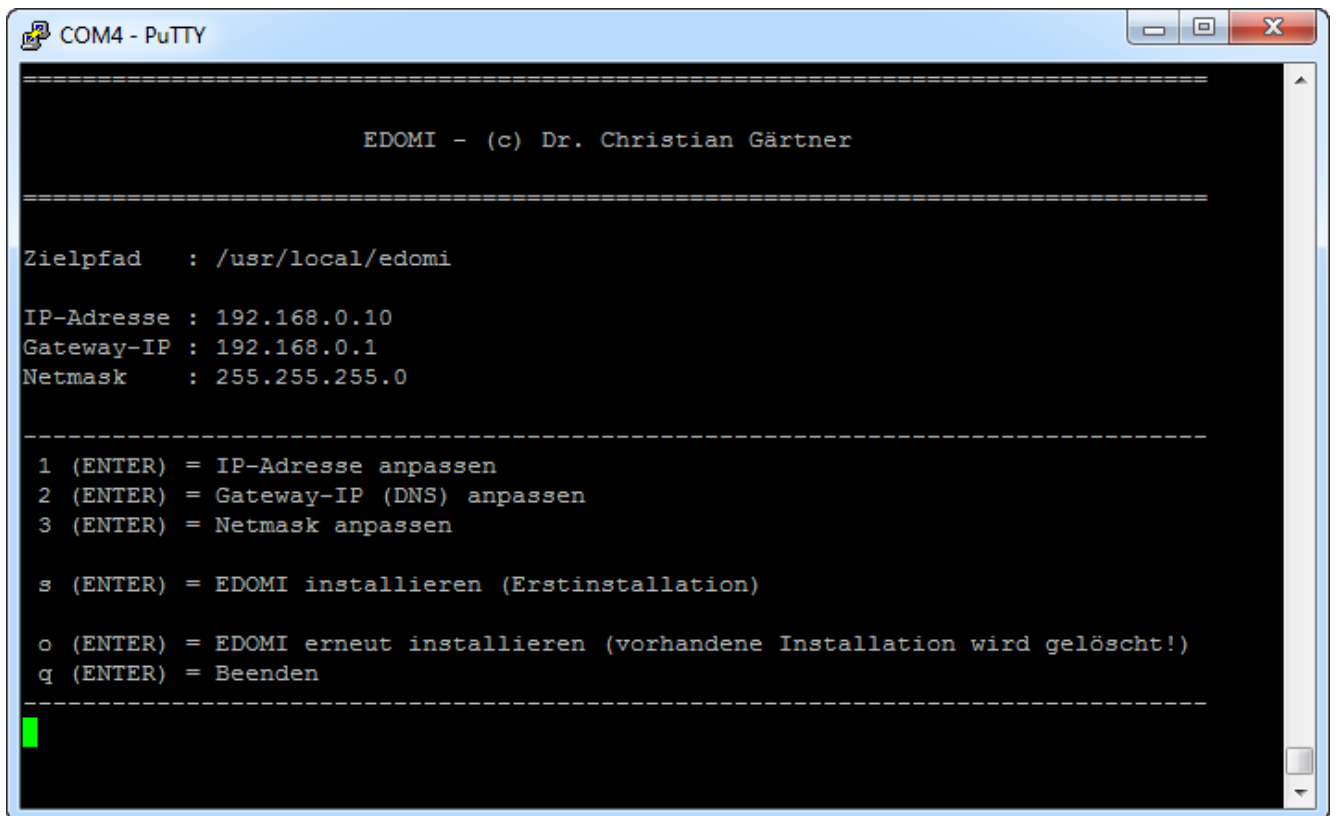
```
./install.sh
```

das Installationscript starten.



```
COM4 - PuTTY
insgesamt 444700
-rwxr-xr-x. 1 root root      2048 29. Nov 2013  boot.cat
-rwxr-xr-x. 1 root root         84 29. Nov 2013  boot.msg
drwxr-xr-x. 3 root root     4096 28. Apr 16:43  edomi
-rwxr-xr-x. 1 root root      334 27. Nov 2013  grub.conf
drwxr-xr-x. 2 root root     4096 28. Apr 16:35  images
-rwxr-xr-x. 1 root root 33392971 27. Nov 2013  initrd.img
-r-xr-xr-x. 1 root root    13639 28. Apr 16:40  ldlinux.sys
-rwxr-xr-x. 1 root root 417333248 28. Apr 16:35  linux.iso
-rwxr-xr-x. 1 root root   165080 29. Nov 2013  memtest
-rwxr-xr-x. 1 root root   151230 29. Nov 2013  splash.jpg
-rwxr-xr-x. 1 root root    1015 28. Apr 16:40  syslinux.cfg
-rwxr-xr-x. 1 root root    2215 29. Nov 2013  TRANS.TBL
-rwxr-xr-x. 1 root root   147996 28. Apr 16:40  vesamenu.c32
-rwxr-xr-x. 1 root root  4128368 29. Nov 2013  vmlinuz
[root@localhost ~]# cd /media/usb/edomi/Install/
[root@localhost Install]# ls -l
insgesamt 28
drwxr-xr-x. 2 root root   4096 28. Apr 16:43  config
drwxr-xr-x. 2 root root   4096 28. Apr 16:43  EDOMI
-rwxr-xr-x. 1 root root  10148 10. Mai 2016  install.sh
drwxr-xr-x. 2 root root   4096 28. Apr 16:43  php
drwxr-xr-x. 3 root root   4096 28. Apr 16:43  rpm
[root@localhost Install]# ./install.sh
```

8. Der erste Kontakt mit EDOMI und wir sehen die **Installationsoptionen**. Auch wenn wir zu diesem Zeitpunkt noch kein Netzwerk definiert haben und vermutlich nicht mal ein Netzwerk angesteckt haben, müssen wir hier die fixe IP, die Gateway bzw. DNS-IP und die IP-Netzwerkmaske angeben.



```
COM4 - PuTTY
=====
EDOMI - (c) Dr. Christian Gärtner
=====

Zielpfad   : /usr/local/edomi
IP-Adresse : 192.168.0.10
Gateway-IP : 192.168.0.1
Netmask    : 255.255.255.0

-----

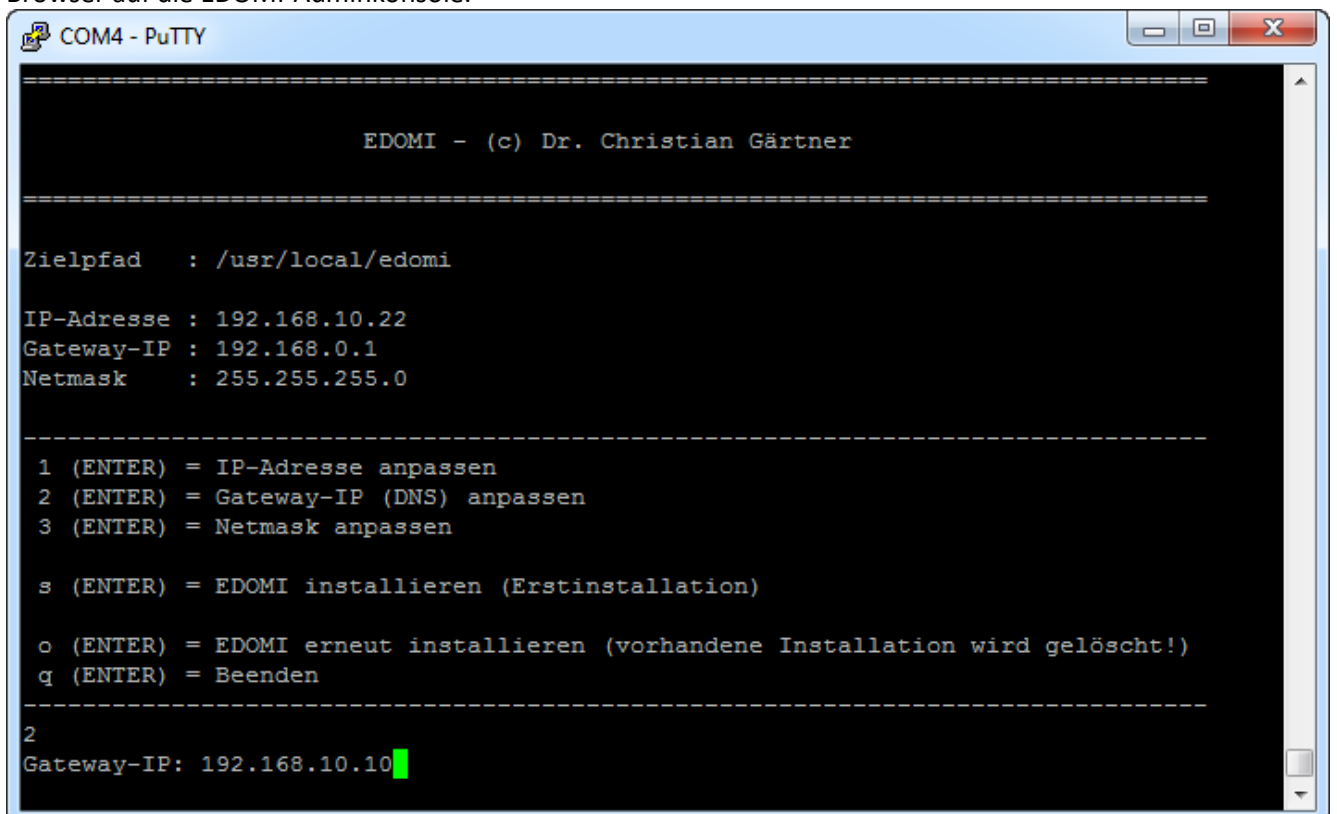
1 (ENTER) = IP-Adresse anpassen
2 (ENTER) = Gateway-IP (DNS) anpassen
3 (ENTER) = Netmask anpassen

s (ENTER) = EDOMI installieren (Erstinstallation)

o (ENTER) = EDOMI erneut installieren (vorhandene Installation wird gelöscht!)
q (ENTER) = Beenden

-----
█
```

9. Mit den Zahlen 1-3 können diese Angaben gemäss deinem Netzwerk zuhause geändert werden. Die IP-Adresse ist später der Zugriff auf das apu2-System via LAN (z.B. SSH) und vorallem natürlich mit dem Browser auf die EDOMI-Adminkonsole.



```
COM4 - PuTTY
=====
EDOMI - (c) Dr. Christian Gärtner
=====

Zielpfad   : /usr/local/edomi
IP-Adresse : 192.168.10.22
Gateway-IP : 192.168.0.1
Netmask    : 255.255.255.0

-----

1 (ENTER) = IP-Adresse anpassen
2 (ENTER) = Gateway-IP (DNS) anpassen
3 (ENTER) = Netmask anpassen

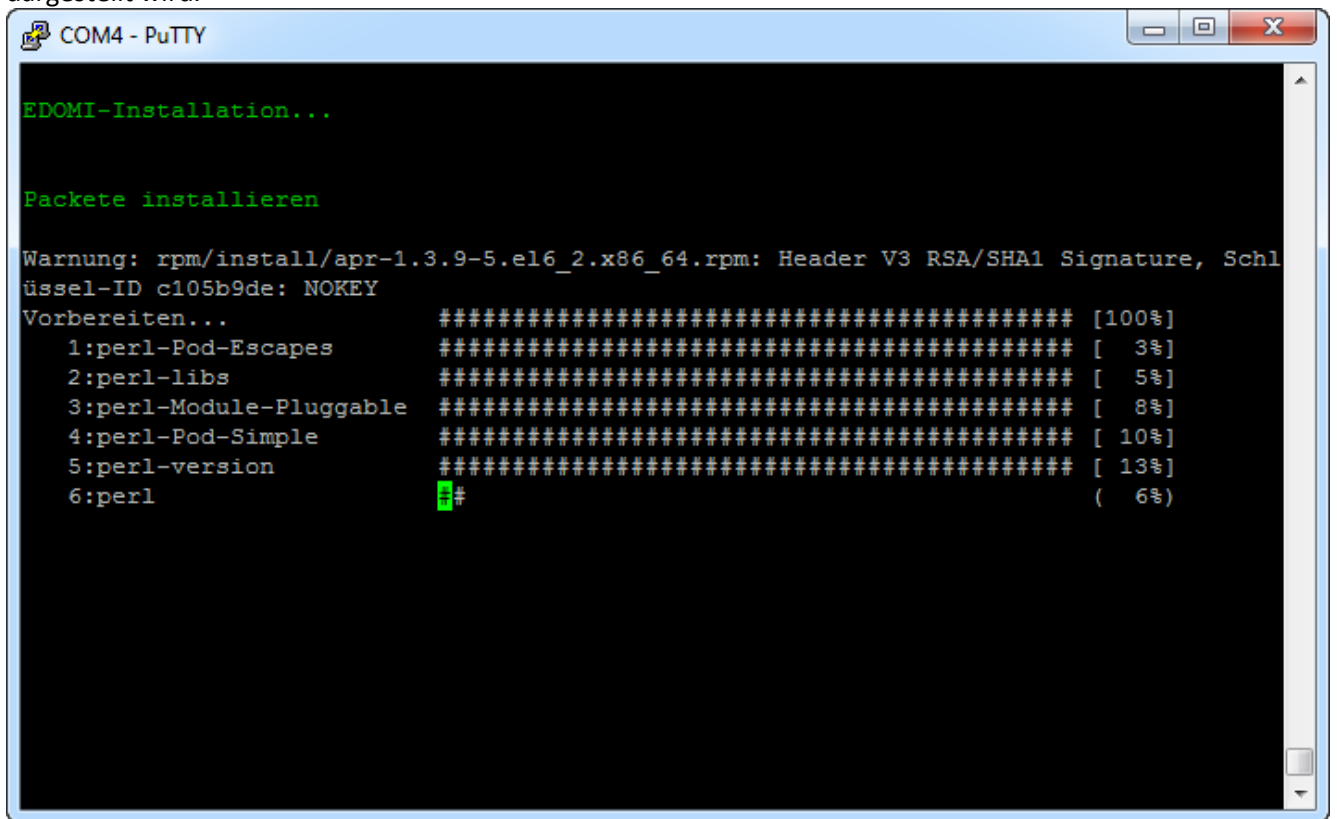
s (ENTER) = EDOMI installieren (Erstinstallation)

o (ENTER) = EDOMI erneut installieren (vorhandene Installation wird gelöscht!)
q (ENTER) = Beenden

-----
2
Gateway-IP: 192.168.10.10 █
```

10. Wurden die Netzwerkangaben definiert, kann mit dem Buchstaben "s" EDOMI das erste Mal (**Erstinstallation**) installiert werden. Ruft man später dieses Installationsscript erneut auf, so würde man "o" (EDOMIT erneut installieren) auswählen. Der Unterschied zwischen s und o ist der, dass bei s alle wichtigen Pakete wie PHP, MySQL

etc. und bei o nur die eigentlichen EDOMI-Dateien installiert werden.
EDOMI wird nun installiert, was (in der für EDOMI typischen liebevollen Übersichtlichkeit) schön dargestellt wird.



```
COM4 - PuTTY
EDOMI-Installation...

Pakete installieren

Warnung: rpm/install/apr-1.3.9-5.el6_2.x86_64.rpm: Header V3 RSA/SHA1 Signature, Schlüssel-ID c105b9de: NOKEY
Vorbereiten...
 1:perl-Pod-Escapes      ##### [100%]
 2:perl-libs            ##### [  3%]
 3:perl-Module-Pluggable ##### [  5%]
 4:perl-Pod-Simple      ##### [  8%]
 5:perl-version         ##### [ 10%]
 6:perl                 ##### [ 13%]
                       ##### [  6%]
#
```

11. Nach einer kurzen Zeit sollte die Installation abgeschlossen sein, was EDOMI auch entsprechend anzeigt. Mit dem Befehl `cd /root` wechseln wir das Verzeichnis, damit wir den USB-Stick wieder unmounten können. Mit dem Befehl `umount /media/usb` wird der Stick ungemounted (ähnlich wie "ausgeworfen"). Wir können den USB-Stick nun sicher ausstecken. Er wird jetzt nicht mehr benötigt (ausser, man möchte EDOMI erneut installieren.) Jetzt wäre es an der Zeit, das apu2-System mit dem LAN zu verbinden (erste LAN-Buchse rechts der seriellen Schnittstelle) Mit dem Befehl `reboot` starten wir das apu2-System erneut und können dann hoffentlich ein erstes Erfolgserlebnis erfahren.

```

COM4 - PuTTY
---fertig---

-----
          OOOOOOOOOOOO  OOOOOOOOO  OOOOOOOOOO  OOOOOOOOOOOOOOOO  O
          O              OO  OO              OO  OO  O  OO  O  O
          O              O  O              O  O  O  O  O  O  O
          OOOOOOOOO  O              O  O              O  O  O  O  O
          O              O  O              O  O  O  O  O  O  O
          O              OO  OO              OO  O  O  O  O  O  O
          OOOOOOOOOOOO  OOOOOOOOO  OOOOOOOOOO  O  O  O  O  O

          EDOMI-Installation abgeschlossen      (c) Dr. Christian Gärtner

-----
Beim nächsten Reboot wird EDOMI automatisch gestartet! Neustart mit: reboot

[root@localhost Install]# umount /media/usb/
umount: /media/usb: device is busy.
(In some cases useful info about processes that use
the device is found by lsof(8) or fuser(1))
[root@localhost Install]# cd /root/
[root@localhost ~]# umount /media/usb/
[root@localhost ~]# reboot

```

12. Nach dem Starten sollte später auch EDOMI automatisch gestartet werden. Der erste grüne EDOMI-Balken zeigt dies bereits an, obwohl es noch nicht ganz soweit ist.

```

COM4 - PuTTY
-lv_swap. Priority:-1 extents:1 across:4046840k SSD
[ OK ]
Betrete nicht-interaktiven Start-Modus
NET: Registered protocol family 10
lo: Disabled Privacy Extensions
Loopback-Schnittstelle hochfahren: [ OK ]
Schnittstelle eth0 hochfahren: ADDRCONF(NETDEV_UP): eth0: link is not ready
Determining if ip address 192.168.10.22 is already in use for device eth0...
igb: eth0 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX/TX
ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
[ OK ]
Mounting filesystems: [ OK ]
Erneutes Triggering für fehlgeschlagene udev-Ereignisse[ OK ]
sshd starten: [ OK ]
ntpd starten: [ OK ]
vsftpd für vsftpd starten:[ OK ]
mysqld starten: [ OK ]
httpd starten: [ OK ]

-----
          EDOMI - (c) Dr. Christian Gärtner
-----

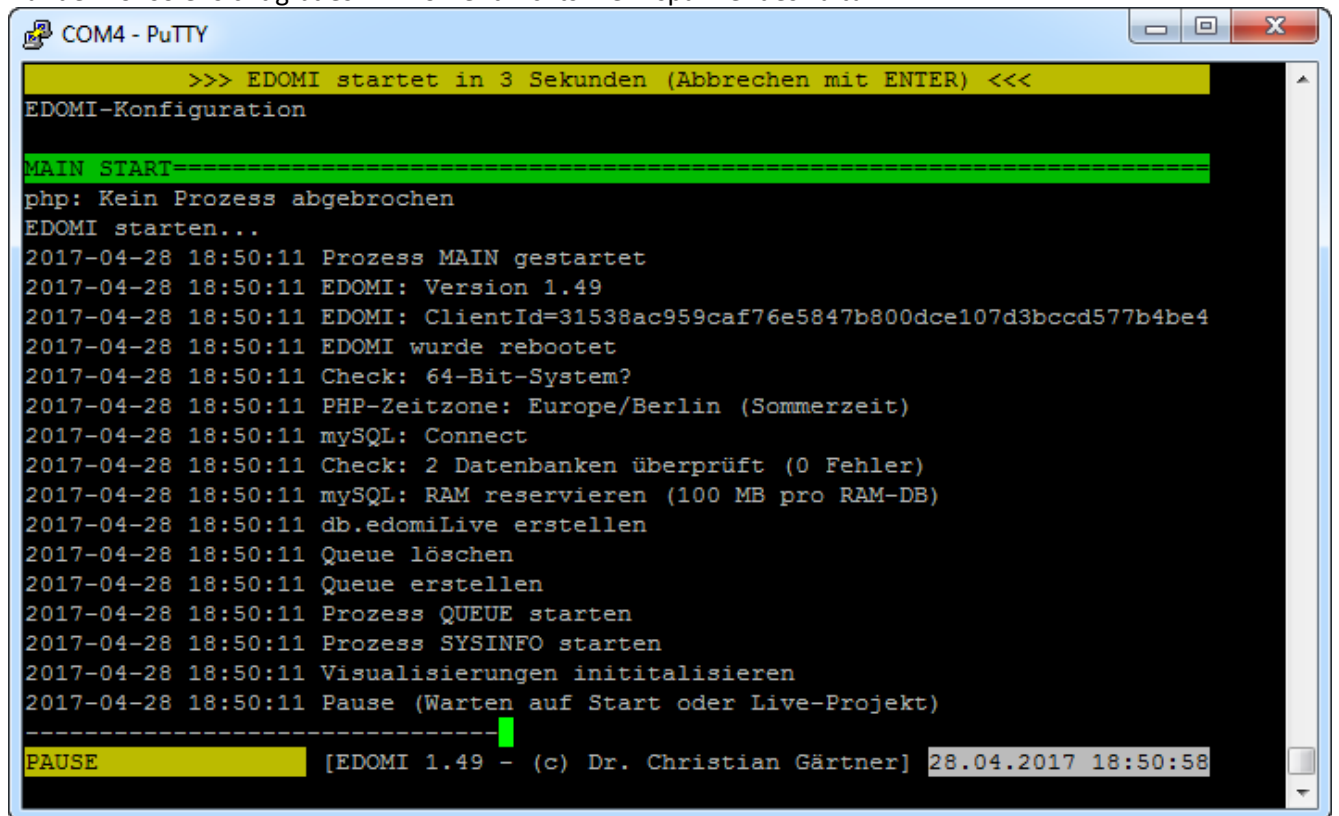
kann Stromsparmmodus nicht (zurück)setzen
mysqld beenden: [ OK ]

```

13. Kurze Zeit später zeigt EDOMI in gelber Schrift eine Meldung an, dass in 3 Sekunden EDOMI gestartet wird. Drückt man in diesem Augenblick ENTER, wird der Start abgebrochen und man befindet sich auf der CentOS Betriebssystem-Ebene. Im Normalfall lässt man aber EDOMI starten. Nach dem Start kommt die evtl. etwas verwirrende Meldung "PAUSE" am Ende der Anzeige. Das hat damit zu tun, dass EDOMI nun zwar gestartet ist, aber noch kein Projekt definiert wurde. Dies ist also

normal.

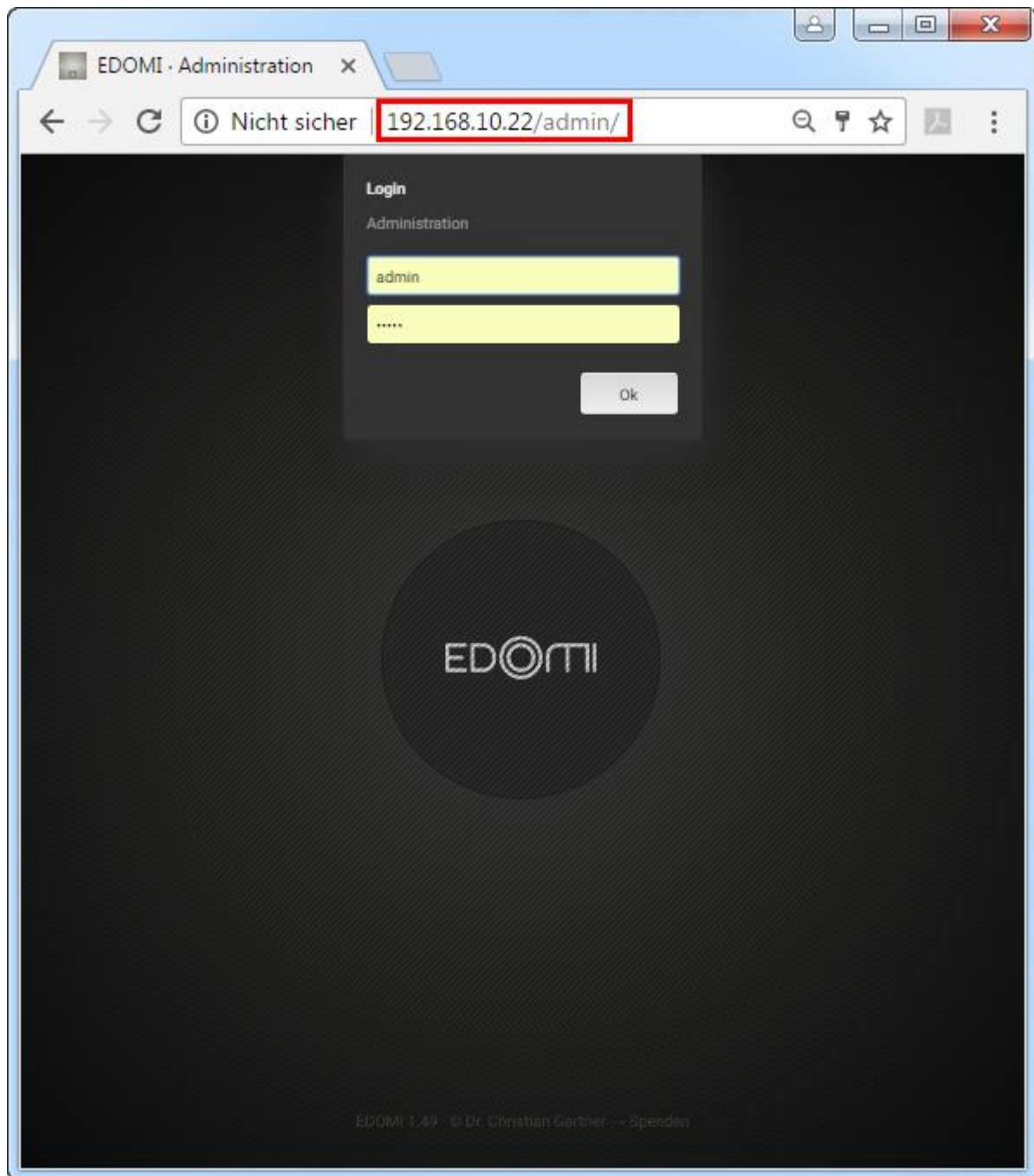
Auf der Konsolensicht gibt es im Moment nichts mehr spannendes zu tun.



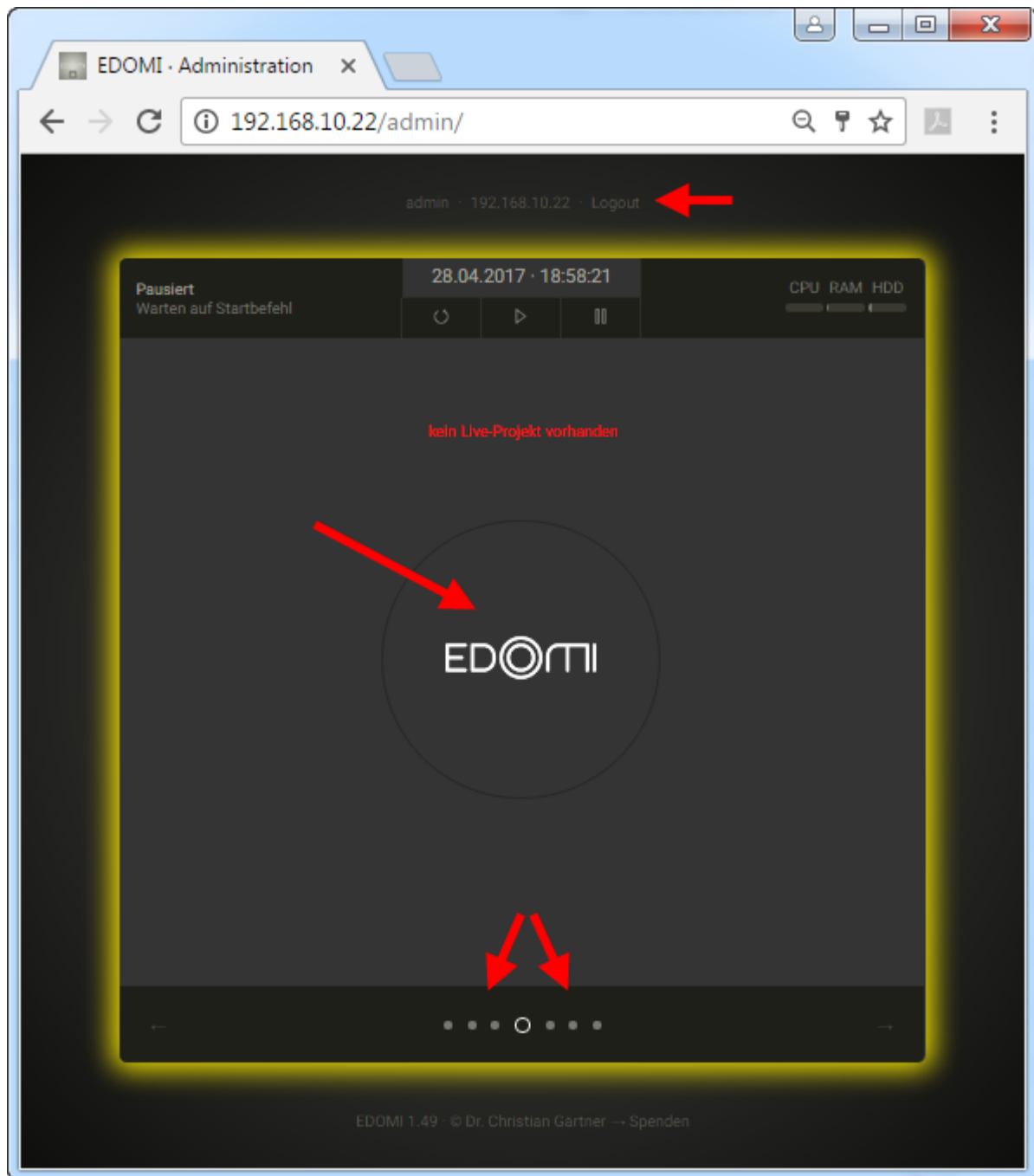
```
>>> EDOMI startet in 3 Sekunden (Abbrechen mit ENTER) <<<
EDOMI-Konfiguration

MAIN START=====
php: Kein Prozess abgebrochen
EDOMI starten...
2017-04-28 18:50:11 Prozess MAIN gestartet
2017-04-28 18:50:11 EDOMI: Version 1.49
2017-04-28 18:50:11 EDOMI: ClientId=31538ac959caf76e5847b800dce107d3bccd577b4be4
2017-04-28 18:50:11 EDOMI wurde rebootet
2017-04-28 18:50:11 Check: 64-Bit-System?
2017-04-28 18:50:11 PHP-Zeitzone: Europe/Berlin (Sommerzeit)
2017-04-28 18:50:11 MySQL: Connect
2017-04-28 18:50:11 Check: 2 Datenbanken überprüft (0 Fehler)
2017-04-28 18:50:11 MySQL: RAM reservieren (100 MB pro RAM-DB)
2017-04-28 18:50:11 db.edomiLive erstellen
2017-04-28 18:50:11 Queue löschen
2017-04-28 18:50:11 Queue erstellen
2017-04-28 18:50:11 Prozess QUEUE starten
2017-04-28 18:50:11 Prozess SYSINFO starten
2017-04-28 18:50:11 Visualisierungen inititalisieren
2017-04-28 18:50:11 Pause (Warten auf Start oder Live-Projekt)
-----
PAUSE [EDOMI 1.49 - (c) Dr. Christian Gärtner] 28.04.2017 18:50:58
```

14. Deshalb wechseln wir jetzt in einen Browser (vorzugsweise Google Chrome oder Safari) und geben als URL die vorher definierte IP ein:
http://192.168.10.22/admin
Haben wir keine Netzwerk-Probleme, sollte die Anmeldemaske der EDOMI-Adminkonsole erscheinen. Mit dem Default-Account **admin** und Passwort **123456** können wir uns anmelden.

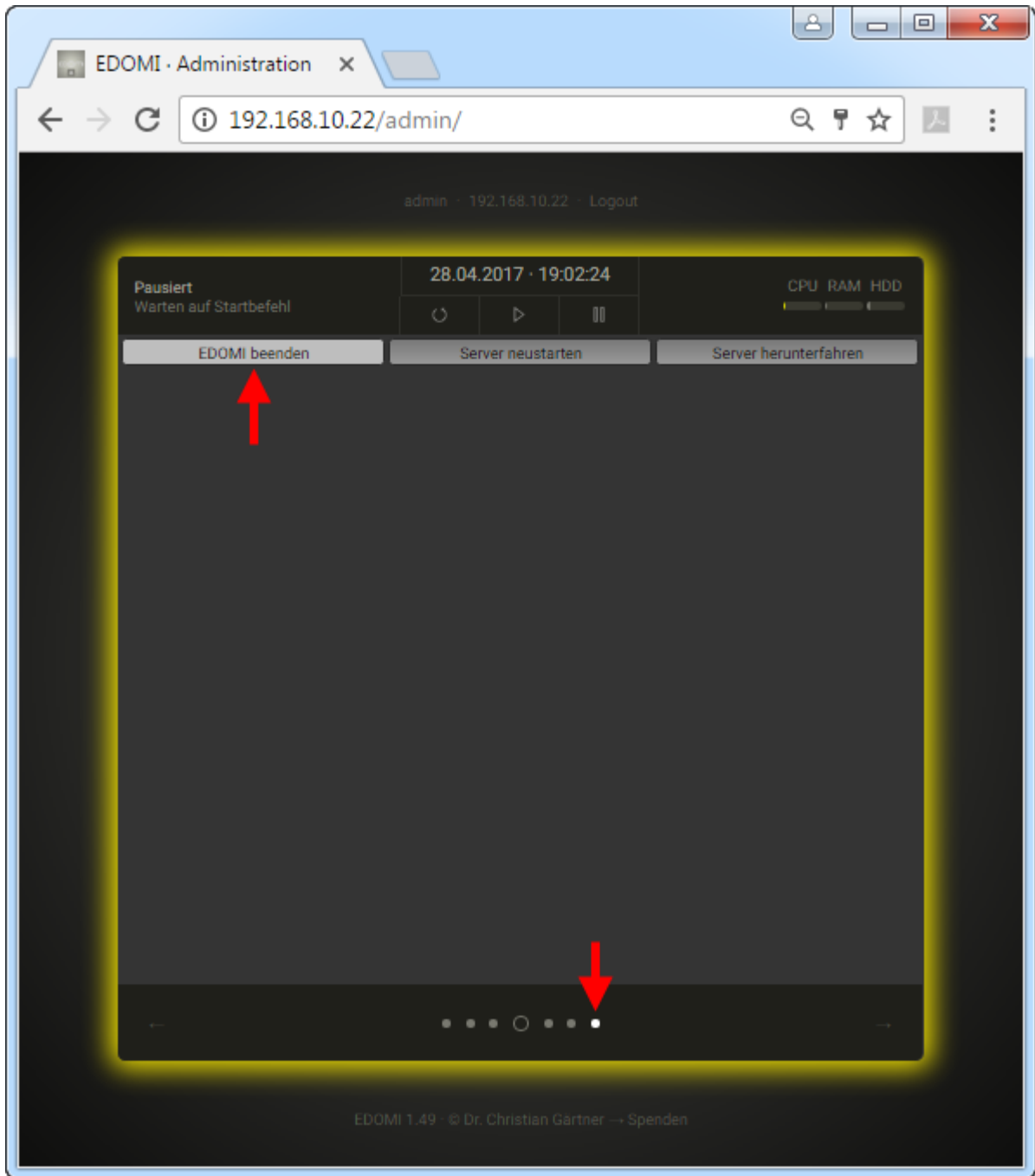


15. Ohne jetzt zu sehr in die Beschreibung von EDOMI einzugehen (dies ist ja nur eine Installationsanleitung), hier ein paar grundlegende Bereiche:
- Oben kann man sich wieder abmelden
 - In der Mitte (innerer Kreis) wird die eigentliche EDOMI-Adminkonsole gestartet
 - Unten kann man sogenannte Widgets aufrufen (jeder Punkt ist eine Seite). Das sind nützliche Anzeigen.

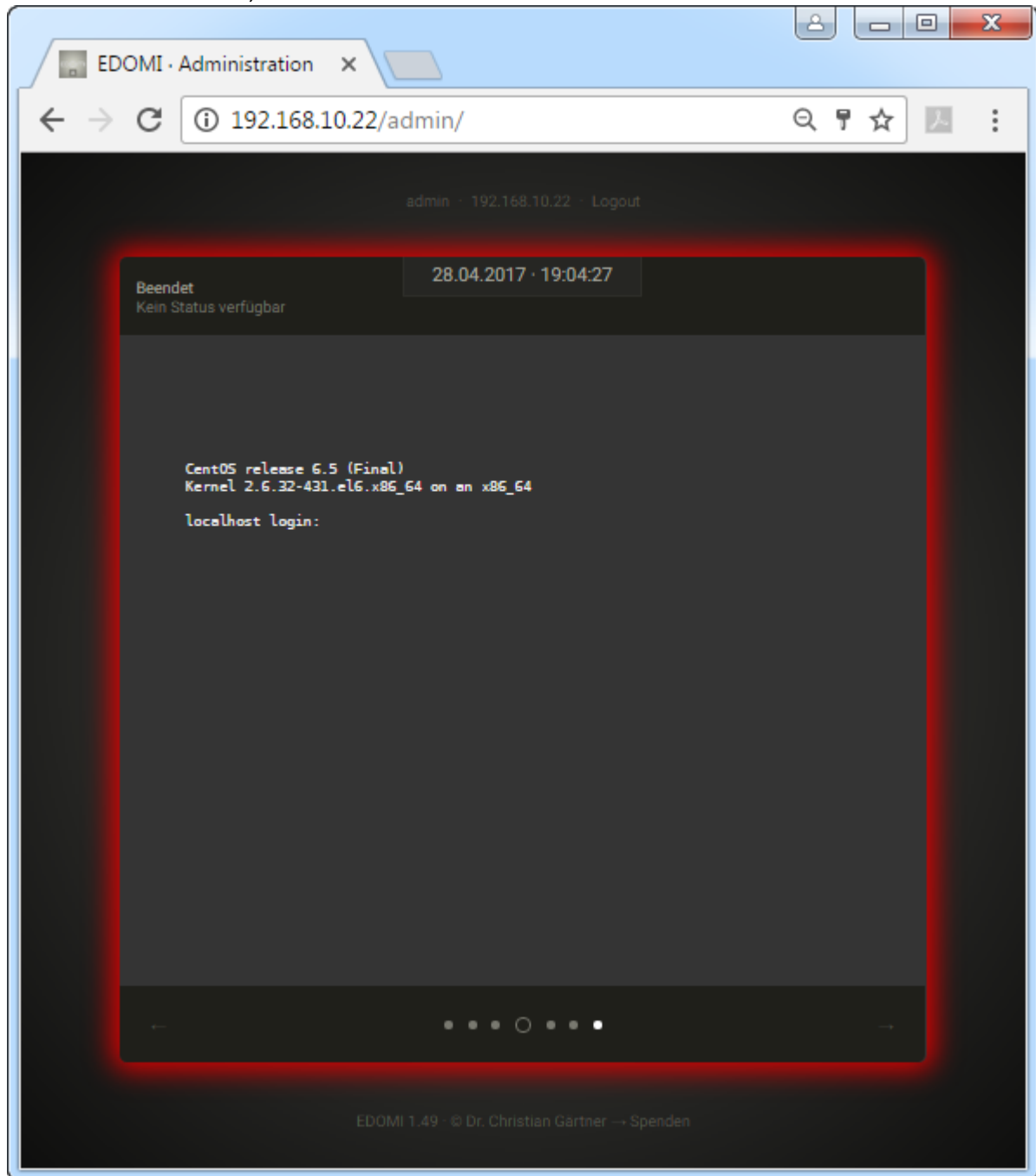


16. Beim letzten Widget-Punkt (rechts) gibt es oben drei Buttons:
- EDOMI beenden (EDOMI wird beendet und man kann auf dem CentOS Command Prompt weiterfahren)
 - Server neustarten (CentOS und EDOMI werden komplett neu gestartet)
 - Server herunterfahren (CentOS und EDOMI werden heruntergefahren.)
- CentOS könnte man auch via SSH-Befehl neustarten (*reboot*) bzw. herunterfahren (*poweroff*), dennoch sollte dies vermieden werden, da EDOMI dies nicht mitbekommt und u.U. nicht sauber beendet wird.
Immer CentOS über diese beiden EDOMI-Befehle neustarten bzw. herunterfahren!

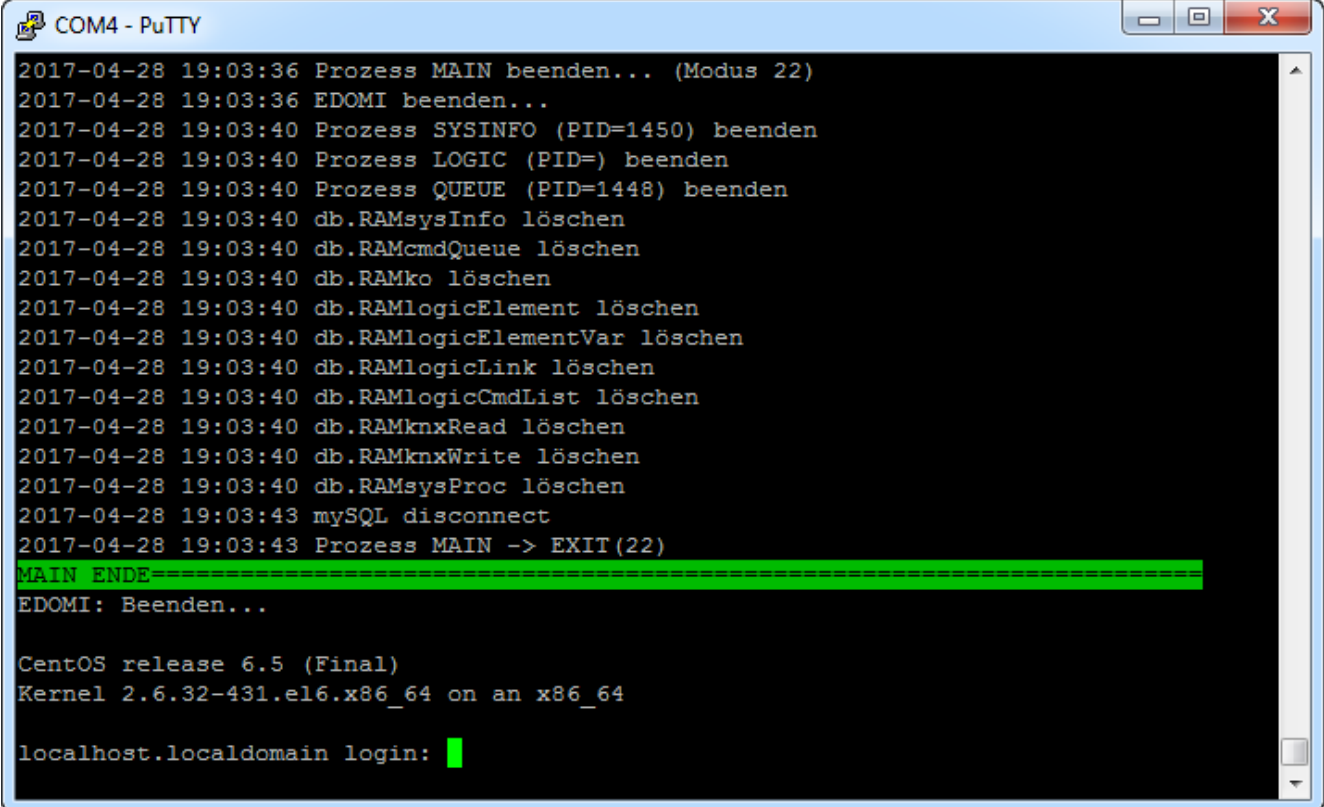
Hinweis: Leider zeigt EDOMI auf einem apu2-System auf dieser Seite keine Konsolensicht an. Auf anderen Systemen wird hier ein Abbild der aktuellen CentOS Konsole angezeigt.



17. Wurde EDOMI beendet, so wechselt die Farbe der Adminkonsole auf rot.



18. Auf der CentOS Konsole (Putty) wird ebenfalls angegeben, dass EDOMI beendet wurde. Zudem erscheint nun wieder der Login Prompt. Man kann sich somit hier mit dem root Account anmelden.



```
COM4 - PuTTY
2017-04-28 19:03:36 Prozess MAIN beenden... (Modus 22)
2017-04-28 19:03:36 EDOMI beenden...
2017-04-28 19:03:40 Prozess SYSINFO (PID=1450) beenden
2017-04-28 19:03:40 Prozess LOGIC (PID=) beenden
2017-04-28 19:03:40 Prozess QUEUE (PID=1448) beenden
2017-04-28 19:03:40 db.RAMsysInfo löschen
2017-04-28 19:03:40 db.RAMcmdQueue löschen
2017-04-28 19:03:40 db.RAMko löschen
2017-04-28 19:03:40 db.RAMlogicElement löschen
2017-04-28 19:03:40 db.RAMlogicElementVar löschen
2017-04-28 19:03:40 db.RAMlogicLink löschen
2017-04-28 19:03:40 db.RAMlogicCmdList löschen
2017-04-28 19:03:40 db.RAMknxRead löschen
2017-04-28 19:03:40 db.RAMknxWrite löschen
2017-04-28 19:03:40 db.RAMsysProc löschen
2017-04-28 19:03:43 mySQL disconnect
2017-04-28 19:03:43 Prozess MAIN -> EXIT(22)
MAIN ENDE=====
EDOMI: Beenden...

CentOS release 6.5 (Final)
Kernel 2.6.32-431.el6.x86_64 on an x86_64

localhost.localdomain login: █
```

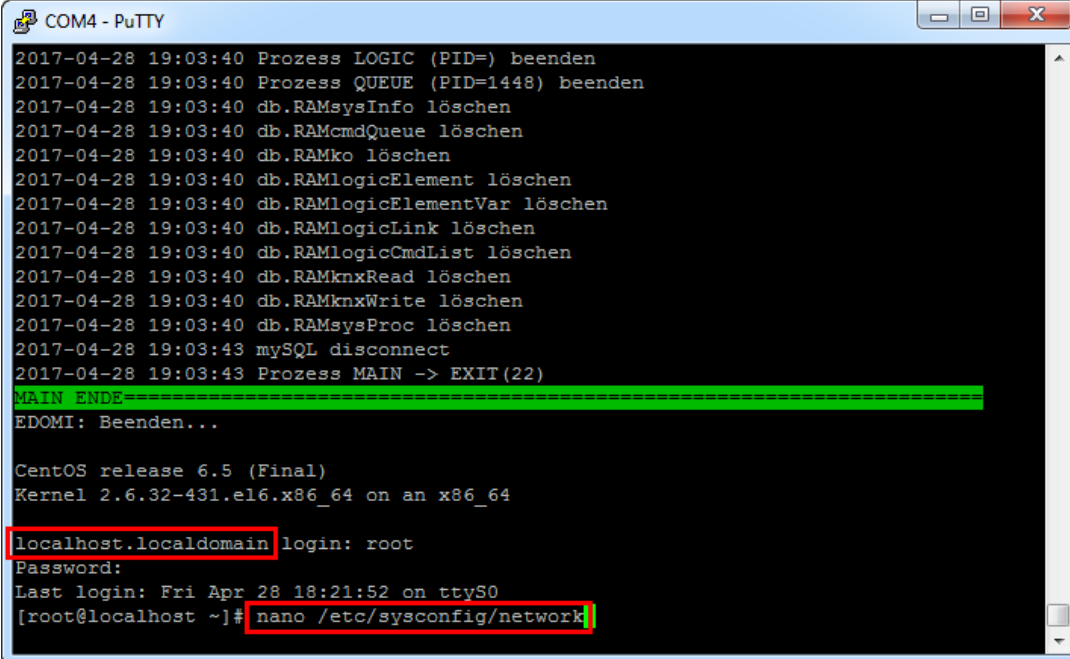
Hinweis: Weitere Informationen zur Bedienung und Konfiguration von EDOMI findest du in der sehr guten integrierten Hilfe (im Browser), sowie natürlich auch im [EDOMI-Supportforum](#).

CentOS Hostname ändern

Manchen wird aufgefallen sein, dass der standardmässige Hostname nach der CentOS Installation localhost.localdomain lautet. (Dies wird auch immer beim Login Prompt angezeigt)

Wer diesen schwerfälligen Namen ändern möchte, kann dies mit folgenden Schritten durchführen:

1. Sich mit dem root-Account am CentOS anmelden.
2. Den Befehl **nano /etc/sysconfig/network** eingeben. Dies startet einen einfachen Texteditor (nano), mit dem die Datei "network" angepasst werden kann.



```
COM4 - PuTTY
2017-04-28 19:03:40 Prozess LOGIC (PID=) beenden
2017-04-28 19:03:40 Prozess QUEUE (PID=1448) beenden
2017-04-28 19:03:40 db.RAMsysInfo löschen
2017-04-28 19:03:40 db.RAMcmdQueue löschen
2017-04-28 19:03:40 db.RAMko löschen
2017-04-28 19:03:40 db.RAMlogicElement löschen
2017-04-28 19:03:40 db.RAMlogicElementVar löschen
2017-04-28 19:03:40 db.RAMlogicLink löschen
2017-04-28 19:03:40 db.RAMlogicCmdList löschen
2017-04-28 19:03:40 db.RAMknxRead löschen
2017-04-28 19:03:40 db.RAMknxWrite löschen
2017-04-28 19:03:40 db.RAMsysProc löschen
2017-04-28 19:03:43 mySQL disconnect
2017-04-28 19:03:43 Prozess MAIN -> EXIT (22)
MAIN ENDE-----
EDOMI: Beenden...

CentOS release 6.5 (Final)
Kernel 2.6.32-431.el6.x86_64 on an x86_64

localhost.localdomain login: root
Password:
Last login: Fri Apr 28 18:21:52 on ttyS0
[root@localhost ~]# nano /etc/sysconfig/network
```

3. Als HOSTNAME anstatt localhost.localdomain einen anderen Namen eingeben (nur ASCII-Buchstaben, Zahlen und der Bindestrich, keine Sonderzeichen)
Mit **CTRL/STRG+O** wird die Datei gespeichert.
Mit **CTRL/STRG+X** wird Nano beendet. (man kann auch beim Schliessen noch speichern)

```
COM4 - PuTTY
GNU nano 2.0.9      Datei: /etc/sysconfig/network      Verändert
NETWORKING=yes
HOSTNAME=edomi-prod
^G Hilfe      ^O Speichern ^R Datei öffn ^Y Seite zurü ^K Ausschneid ^C Cursor
^X Beenden    ^J Ausrichten ^W Wo ist   ^V Seite vor ^U Ausschn. r ^T Rechtschr.
```

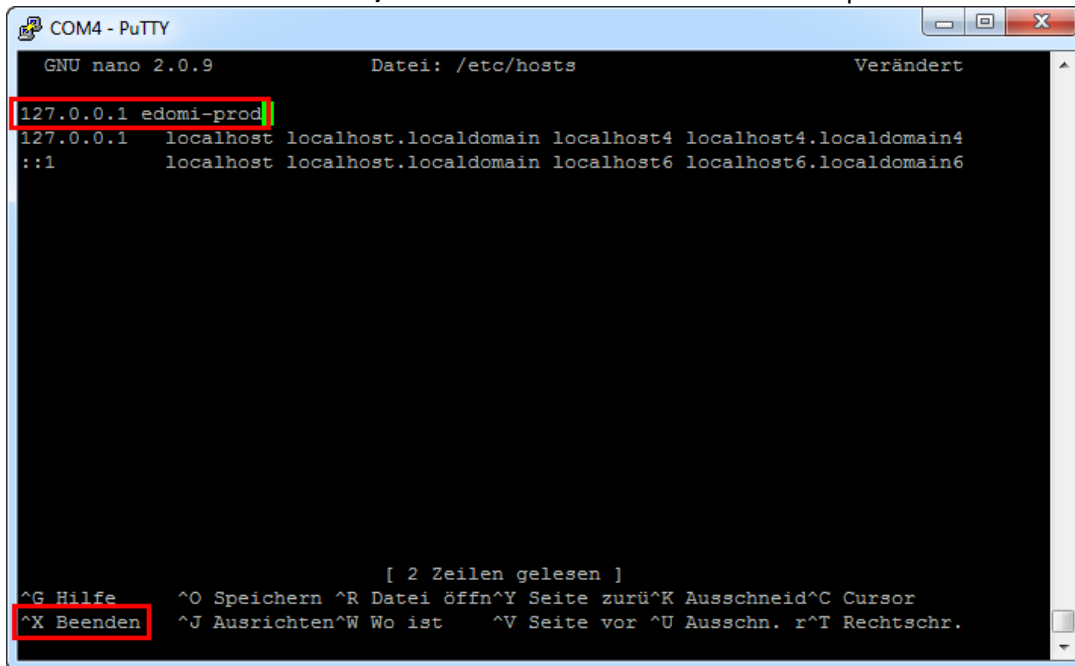
4. Sinnvollerweise (nicht zwingend) passen wir auch noch die hosts-Datei mit folgendem Befehl an:
nano /etc/hosts

```
COM4 - PuTTY
GNU nano 2.0.9      Datei: /etc/sysconfig/network
NETWORKING=yes
HOSTNAME=edomi-prod

[ 2 Zeilen geschrieben ]
[root@localhost ~]# nano /etc/hosts
```

5. Hier fügen wir zuoberst mit der RETURN-Taste eine neue Zeile ein und geben das folgende ein:
127.0.0.1 NEUER_NAME
(NEUER_NAME muss natürlich mit deinem neuen Host-Namen ersetzt werden. Zwischen der Localhost IP und dem Namen muss mindestens ein Leerzeichen stehen.)

Anschliessend wieder mit **CTRL/STRG+X** nano beenden und die Datei speichern.



```
COM4 - PuTTY
GNU nano 2.0.9      Datei: /etc/hosts      Verändert
127.0.0.1 edomi-prod
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

[ 2 Zeilen gelesen ]
^G Hilfe      ^O Speichern ^R Datei öffn^Y Seite zurü^K Ausschneid^C Cursor
^X Beenden    ^J Ausrichten^W Wo ist      ^V Seite vor ^U Ausschn. r^T Rechtschr.
```