

# TGGs Debian 13 (Trixie) KDE Plasma

## Stand: 25.07.2025

### Bitte lesen – Wichtige Informationen zum System

## Minimale KDE-Installation

Dieses System basiert auf einer bewusst schlanken KDE-Plasma-Grundinstallation. Es sind nur die absolut notwendigen Programme vorinstalliert, um dir maximale Freiheit zu bieten. Du entscheidest, was du brauchst – keine Vorgaben, keine Bloatware.

## ISO Erklärung

Die NVIDIA Karten treiben mich momentan ein wenig zur Verzweiflung, weshalb ich mittlerweile einige ISO Varianten anbiete.

 <a href="#">20072025_TGG_Debian_Trixie_KDE.iso</a>	2025-07-20 14:57	2.3G
 <a href="#">23072025_TGG_DEBIAN_Trixie_KDE_NVIDIACLOSED.iso</a>	2025-07-24 19:05	2.8G
 <a href="#">23072025_TGG_DEBIAN_Trixie_KDE_NVIDIAOPEN.iso</a>	2025-07-24 06:36	2.7G
 <a href="#">25072025_TGG_Debian13_KDE_UEFI.iso</a>	2025-07-25 17:26	970M

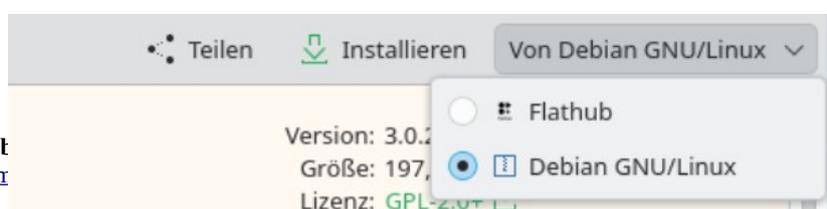
- **Erste Datei:** Bootmodus individuell, **kein** NVIDIA „nomodeset“ Parameter für „Problemgrafikkarten“. (Problemgrafikkarten laufen nicht)
- **...NVIDIACLOSED.iso:** Vorinstallierter NVIDIA 570 Treiber (closed Source)
- **...NVIDIAOPEN.iso:** Vorinstallierter NVIDIAOPEN 570 Treiber (open Kernelmodul) für Kernel 6.15 und neuer (und auch älter).
- **...UEFI.iso:** Bootmodus UEFI – mit vorausgewähltem „nomodeset“ im Bootmenü. (Problemgrafikkarten sollten im Live-Mode laufen, um zumindest das System installieren zu können).

## /etc/apt/sources.list (Debian Paketquellen)

Von den Debian Paketquellen her, sind alle Zweige bereits eingebunden. Diese umfassen: **main**, **contrib**, **non-free** und **non-free-firmware**. Eine Bearbeitung der `/etc/apt/sources.list` ist **nicht** mehr nötig.

## Flatpak/Flathub integriert

Die Discover Paketverwaltung beinhaltet auch bereits die Flatpakintegration. Ihr könnt somit via Discover auch Flatpak-Pakete installieren. In der Regel habt ihr oben rechts ein Dropdownmenü. Über dieses Menü könnt ihr entscheiden, ob das gewählte Paket von den Debian-Paketquellen, oder vom Flathub kommt. **Tipp:** Mit Flatpak sparsam umgehen!



## Programme einfach nachinstallieren

Über den Desktop-Button „**Programme installieren**“ kannst du gängige Anwendungen (Browser, Office, Multimedia usw.) schnell und unkompliziert hinzufügen. Dies richtet sich vor allem an Anfänger, die ggf. nach Standardprogrammen suchen und vielleicht einen alternativen Internetbrowser installieren wollen, ohne an den Paketquellen etwas ändern zu müssen.



Der Button startet das Installer-Script, das als normales Bashscript (Konsole) vorliegt. Die Bedienung erfolgt mit den Cursoratsten & Return → zur Auswahl des Menüpunkts.

Über dieses Menü könnt ihr Software installieren, die nicht in den Debian-Paketquellen vorhanden sind. Es werden Fremdquellen eingebunden. Diese liegen in /etc/apt/sources.list.d.

```
TGG Programmauswahl
Wähle ein Programm zur Installation aus:
 1 Brave Browser (Brave Repository)
 2 Mozilla Firefox (Mozilla Repository)
 3 LibreWolf Browser (Datenschutzfreundlich, via extrepo)
 4 Thunderbird E-Mail-Client
 5 GIMP Bildbearbeitung
 6 OBS Studio (Streaming/Recording)
 7 Kdenlive Video Editor
 8 Steam installieren (Spieleplattform)
 9 Lutris installieren (Spieleplattform)
10 LibreOffice (Office-Suite)
11 ONLYOFFICE (Office-Suite)
12 Alle Skripte aktualisieren (tgg_script_updater.sh)

<Ok>                                <Abbrechen>
```

Alternativ steht dir die KDE-Softwareverwaltung „**Discover**“ zur Verfügung, über die du zusätzliche Programme grafisch suchen und installieren kannst. Hier hast du Zugriff auf so gut wie alle verfügbaren Pakete von Debian 13 & Flatpak, jedoch nicht auf die Programme aus Fremdquellen. (Wie zum Beispiel den Brave Browser, der aus einer „debianfremden“ Quelle

kommt und nur über „**Programme installieren**“ am Desktop oder per Konsole - **program\_installer.sh** - installiert werden kann.)

**WICHTIG:** Es ist **KEIN Browser** vorinstalliert! Den Firefox ESR kannst du -wenn gewünscht- über Discover installieren. Es ist allerdings ein Extended Support Release und somit nicht die aktuellste Version! Die restlichen Browser mit dem Button „**Programme installieren**“ am Desktop. Google Chrome habe ich aus Datenschutzgründen weggelassen.

Menüpunkt 12: **Alle Skripte aktualisieren** → prüft ob auf meinem Codeberg.org Repository ([https://codeberg.org/techgamegeek/TGG\\_Debian\\_Trixie](https://codeberg.org/techgamegeek/TGG_Debian_Trixie)) neuere Versionen der Skripts, die unter /usr/local/bin liegen, vorhanden sind und aktualisiert sie gegebenenfalls. Auf diesem Weg kann ich zum Beispiel eventuell Probleme mit den Skripts beheben. Ihr startet (wenn ihr wollt) einfach das Update aller meiner Skripts mit Menüpunkt 12.

## Steam / Lutris

Das i386-Archiv (**dpkg --add-architecture i386**) wird aktiviert, das unbedingt für den Betrieb und die Installation dieser Plattformen nötig ist bzw. wird die ausgewählte Spieleplattform installiert.

## NVIDIA (**only**) Wartungsmenü

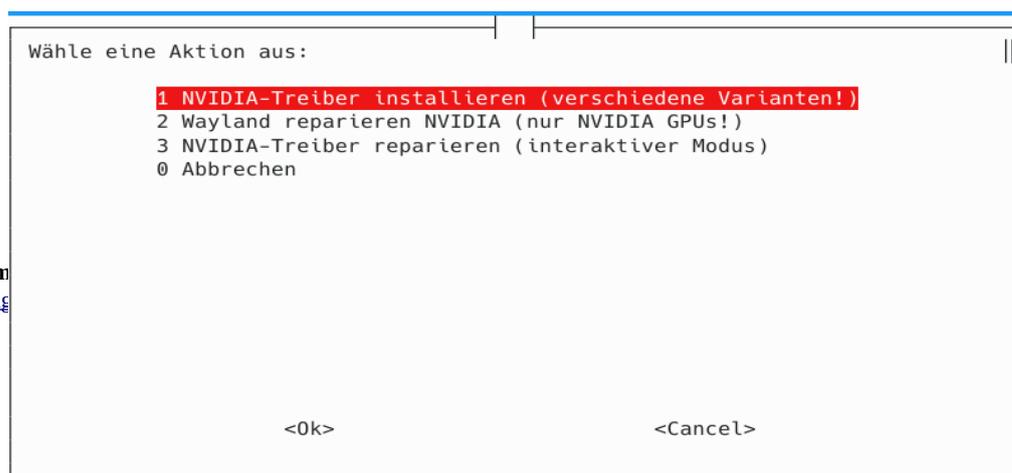
Auf dem Desktop findet ihr neben der Programminstallation auch noch ein Symbol (Link) zum NVIDIA-Wartungsmenü. Dieses Symbol ist mit dem Skript „**repair\_installer.sh**“ verknüpft. Wenn ihr nicht auf die grafische Oberfläche kommt (Treiberproblem, Waylandfehler etc.), dann kann es von der Konsole aus aufgerufen werden, indem ihr einfach **repair\_installer.sh** eingibt.

**Tip:** Wenn ihr nur einen blinkenden Cursor nach dem Boot seht und eure Desktopoberfläche nicht starten will, drückt **STRG + ALT + F1 bis F6** um eine Konsole zu öffnen. Jede F-Taste ist eine eigene Konsole.

Bootet eure Desktopumgebung seht ihr folgendes Symbol:



Bei Doppelklick darauf öffnet sich ein spezielles Wartungsmenü für NVIDIA Grafikkarten. (Das seht ihr übrigens auch, wenn ihr den **repair\_installer.sh** von der Konsole startet.



## Details zu Option „NVIDIA-Treiber installieren.“

**Wichtig:** Nachdem ich nun auch einen Kernel-Installer hinzugefügt habe mit dem ihr sehr aktuelle Kernelversionen (abgeleitet vom Vanilla-Kernel – von Kernel.org) installieren könnt: Ihr dürft **KEINEN CUDA** oder **NVIDIA-DRIVER** installieren, der das proprietäre Kernelmodul verwendet. Sobald ihr ein Kernel Version 6.15 (oder höher) installiert, müsst ihr per **install\_nvidia.sh** den installierten Treiber deinstallieren und dann den Treiber mit dem Opensource-Kernelmodul installieren. (Nochmals: NICHT CUDA und auch nicht **nvidia-driver** aus den Debian Repositories). Es gibt aktuell eine Inkompatibilität, die aufgrund von GPL Vorgaben des Kernels auftritt, die nun strikter durchgesetzt werden. **Nur GPL-kompatibler Code** darf Kernel-nahe Module verwenden.

Das Opensource-Kernelmodul unterstützt nicht alle (aber sehr viele NVIDIA GPUs): Siehe <https://github.com/NVIDIA/open-gpu-kernel-modules?tab=readme-ov-file#compatible-gpus>

**Kurz:** Wenn euer Kernel  $\geq V6.15$  ist, oder ihr vor habt (m)einen aktuellen Kernel zu installieren (momentan 6.15) » Bitte **vorher** beim Installieren des NVIDIA Drivers (per Menü – siehe oben), den Treiber mit dem Opensource-Kernelmodul wählen.

Der Dialog sieht so aus:

```
=====
NVIDIA-Treiberinstallation - Auswahlmöglichkeit
=====

1) Installation über das CUDA-Repository (APT-basiert, stable)
  > Einfache Installation über Paketverwaltung
  > Kompatibel mit Kernel 6.1 (Debian Stable)
  △ Keine Unterstützung für Kernel ≥ 6.15!
    Der proprietäre Treiber erfüllt nicht die GPL-Anforderungen.

2) Manuelle .run-Installation (mit Open Kernel Modules - OKM)
  > Kompatibel mit Kernel 6.15+
  > Offizieller NVIDIA-Support (ab v515+) - Open-Source-Module

      ==STRG + C für Abbruch ==

Welche Methode möchtest du verwenden? (1/2): █
```

Hier **unbedingt 2** wählen wenn ihr vor habt Kernel 6.15+ zu installieren!

Ihr verliert damit NICHT die Funktionalitäten eurer NVIDIA Karte, da diese den sogenannten Userspace betreffen und weiterhin per NVIDIA (proprietär) abgedeckt sind.

#### **Ergebnis:**

- NVIDIA-Treiber installiert.
- Nouveau ist deaktiviert.
- System ist bereit für CUDA-Anwendungen, OBS mit NVENC, Steam mit Proton usw.
- dkms sorgt dafür, dass der NVIDIA-Treiber bei Kernel-Updates automatisch neu gebaut wird.

#### **Details zu Option „Wayland reparieren...“**

Das Skript hilft dabei, dass NVIDIA-Grafikkarten richtig mit dem modernen Wayland-System funktionieren. Das Skript müsst ihr von einer Konsole aus ausführen, da es nur dann zum Einsatz kommt, wenn ihr eine NVIDIA Karte verwendet und nicht auf eure Desktopoberfläche kommt.

## **Symptome einer fehlerhaften Wayland-Konfiguration & NVIDIA:**

### → **Schwarzer Bildschirm nach Login / kein Desktop (Wayland startet nicht)**

- Der Loginmanager (z. B. GDM, SDDM) erscheint, aber nach dem Einloggen in eine Wayland-Session bleibt der Bildschirm schwarz.
- Oft kehrt man sofort zum Loginbildschirm zurück (Login-Loop).
- Ursache: **nvidia-drm modeset=1** fehlt → KMS (Kernel Mode Setting) ist nicht aktiv, was Wayland zwingend erwartet. (Das Skript erledigt das Setzen dieser Einstellung).

### → **Wayland-Sitzung ist im Loginmanager gar nicht auswählbar**

- Nur "Plasma (X11)" oder "GNOME auf X11" verfügbar
- "GNOME (Wayland)" fehlt komplett
- Ursache: Das System erkennt, dass NVIDIA nicht Wayland-kompatibel konfiguriert ist → Option wird **versteckt** (z. B. durch GDM)

## **Details zu Option NVIDIA „Treiber reparieren (interaktiver Modus)“**

Diese Option entfernt alte Treiber und hilft dir, den richtigen NVIDIA-Treiber oder Nouveau neu zu installieren – Schritt für Schritt mit Auswahlmenü (interaktiv).

**Wichtig:** Auch hier gilt → Wenn ihr ein aktuelles Kernel installieren wollt (z.B. 6.15+) und eine NVIDIA Grafikkarte habt, müsst ihr darauf achten, das Opensource-Kernelmodul von NVIDIA zu installieren. Dies könnt ihr mit dem Skript „**install-nvidia.sh**“ erledigen, oder einfach im NVIDIA Wartungsmenü → Auswahl des ersten Menüpunktes. Bitte in dem Fall hier **nicht mehr weitermachen**. Details zum Opensource-Kernelmodul siehe bitte auf den vorhergehenden Seiten.

Du bekommst ein **Menü mit 3 Auswahlmöglichkeiten:**

- **Aktueller Treiber von NVIDIA** (inkl. CUDA, für neue Karten)
- **Debian-Treiber** (stabil, vom System empfohlen)
- **Freier Nouveau-Treiber** (wenn du keinen proprietären willst)

## **Wann ist diese Option sinnvoll?**

- Du hast einen schwarzen Bildschirm oder kommst nicht mehr in den Desktop nach der Treiberinstallation.
- Du willst zwischen Treibervarianten wechseln (z. B. von Debian-Treiber zu offiziellem NVIDIA-Treiber oder zu Nouveau).

- Nach einem System- oder Kernel-Update funktioniert der NVIDIA-Treiber nicht mehr richtig.
- Du möchtest den freien Nouveau-Treiber testen oder zu ihm zurückkehren.
- Dein System ist **beschädigt** durch alte, manuelle oder fehlerhafte Treiberinstallationen.
- Du willst eine **saubere Neuinstallation** des NVIDIA-Treibers mit klarer Auswahl und ohne Reste.
- Du bist dir **nicht sicher, welcher Treiber am besten passt** – das Menü gibt dir eine einfache Auswahl.

## Terminal-Hinweis & Zusatzwerkzeuge

Beim Öffnen jedes Terminals (Konsole) erhältst du einen Hinweistext zu hilfreichen Werkzeugen:

### Programme-Installations-Menü

– Wie oben beschrieben – für einfache Nachinstallation gängiger Anwendungen. Hiervon liegt auch der Link am Desktop. In der Konsole tippt man einfach den Namen des Scripts ein:

**program\_installer.sh.**

```

=====
[] Programme zur Installation & Wartung verfügbar:
📦 Programm-Installations-Menü:
  program_installer.sh      → Standardprogramme installieren

🔧 NVIDIA-Servicemenü :
  repair_installer.sh      → Diverse Tools zur Reparatur

🔧 Kernel-Wartung:
  kernel_menu.sh           → Kernel installieren/deinstallieren
                           Vanillakernel von kernel.org

Tipp: Alle Skripte befinden sich unter /usr/local/bin
=====

```

### Servicemenü

Wie bereits erwähnt richtet sich dieses Menü **ausschließlich** an NVIDIA Kartennutzer.

### Kernel-Wartung

Hier habt ihr die Möglichkeit Kernel zu deinstallieren bzw. (vor allem) aktuelle Kernel – ich kompiliere sie auf Basis der .config des Debiankernel für x86\_64 – Quellcode von kernel.org, Da es so wichtig ist, möchte ich unbedingt nochmals darauf hinweisen, dass ihr keine Kernelversion 6.15+ verwenden könnt, wenn ihr das proprietäre Kernelmodul von NVIDIA verwendet. Der Bau der nötigen Module schlägt fehl! Bevor ihr ein Kernel 6.15+ installiert, müsst ihr entweder den Nouveau Treiber verwenden, oder die NVIDIA-Open-Kernelmodule!

Im Kernel selbst, habe ich ein paar Optimierungen vorgenommen. Ähnlich wie beim Liquorix-Kernel und einige fix eingebaute Komponenten als Modul geladen. Inwiefern diese Optimierungen tatsächlich etwas bringen, steht in den Sternen. Angeblich erhält man eine etwas schneller reagierendes System. Das muss nicht unbedingt heißen, dass ihr massive Performanceschübe bemerkt (zumeist merkt man gar nichts).

Option	Status	Bedeutung
CONFIG_HZ=1000	✓	1000Hz Tickrate = bessere Reaktionszeit
CONFIG_PREEMPT=y	✓	Volle Kernel-Preemption
CONFIG_PREEMPT_DYNAMIC=y	✓	Umschaltbar zur Laufzeit
CONFIG_NO_HZ_IDLE=y	✓	Tickless-Idle für geringere Latenz
CONFIG_SCHED_AUTOGROUP=y	✓	Bessere Desktop-Reaktionsfähigkeit
CONFIG_IOSCHED_BFQ=y	✓	I/O-Scheduler „BFQ“ für Spiele & Desktops
CONFIG_CPU_FREQ_DEFAULT_GOV_SC	✓	Dynamischer CPU-Governor für bessere Balance
HEDUTIL=y		
CONFIG_UCLAMP_TASK=y	✓	Scheduling-Feintuning für Spiele/Prozesse
CONFIG_HIGH_RES_TIMERS=y	✓	Präzisere Zeitsteuerung
CONFIG_IRQ_FORCED_THREADING=y	✓	Entlastet IRQs – relevant für Audio & Spiele

## Und wenn ich nicht mehr auf den Desktop komme?

Die Reparatur-Tools sind bewusst als Terminalprogramme gehalten – aus gutem Grund:

- Es kann vorkommen, dass die Desktopoberfläche nicht mehr startet (z. B. nur noch ein blinkender Cursor erscheint).
- In diesem Fall kannst du jederzeit ein Terminal (TTY) öffnen:  
*STRG + ALT + F1 - F6*.
- Jede dieser Tasten öffnet ein eigenes Terminal (genannt TTY), unabhängig von der grafischen Oberfläche.
- Dort kannst du das Servicemenü starten, wie es dir beim Öffnen der Konsole im Text angezeigt wird.

Alle Reparatur- und Hilfsprogramme liegen unter: `/usr/local/bin`

Du musst nicht extra in dieses Verzeichnis wechseln und kannst die Reparatur-Tools direkt ausführen.

Falls es nicht klappt, kann jedoch in das Verzeichnis gewechselt werden, um die Befehle danach von dort auszuführen.

## **cd /usr/local/bin**

Generell kann man festhalten, dass Probleme mit der grafischen Oberfläche in 99% der Fälle nur mit NVIDIA Grafikkarten auftreten. Das ist nicht negativ gemeint, ist jedoch dem Umstand geschuldet, dass der NVIDIA-Treiber noch immer größtenteils auf geschlossenem Quellcode basiert und nachinstalliert werden muss, wenn man eine NVIDIA-Karte mit optimaler Geschwindigkeit unter Linux betreiben will.

Treiber von AMD-GPUs sind „Opensource“ und aufgrund dessen in den Linux-Kernel integriert. Das heißt soviel, dass AMD Grafikkarten ohne weitere Aktionen, sofort nach der Installation unterstützt werden. Für den Anwender heißt das: AMD Grafikkarte & Linuxdistribution = wenig bis gar keine Probleme oder Zusatzinstallationen. Auch Wayland läuft mit AMD-GPUs weit stabiler und besser.

## **Automatische Laufwerkseinbindung**

Dieses System bindet Laufwerke automatisch ein. Es ist **kein Eintrag** in die Datei `/etc/fstab` nötig.

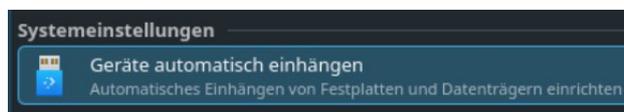
Ich realisiere das mittels dem Programm **udisks2** und Polkit (nur als Information und nicht wichtig).

In KDE Plasma könnt ihr aufgrund dessen einfach auf vorhandene Laufwerke klicken, die dann (nach dem Klick) automatisch in das System eingebunden werden (im Dateibrowser Dolphin).

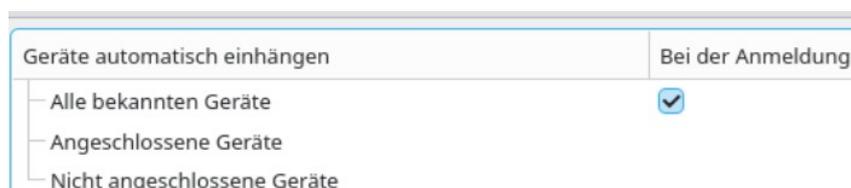
## **Steam Nutzer**

Steam erwartet, dass Bibliotheksfestplatten (=dort wo eure Spiele gespeichert sind) bei Systemstart zur Verfügung stehen. Aus diesem Grund müsst ihr unter KDE – Plasma eine Einstellung anpassen.

1.) Aufruf der „**automatischen Geräteeinbindung**“:



2.) Anhaken von „**Alle bekannten Geräte**“ → Bei der Anmeldung:



3.) Unten rechts auf „Anwenden“ klicken & dann das System neu starten.

## Allgemeines

In der Datei `/etc/apt/sources.list` sind die Debian Paketquellen hinterlegt. Die Backports von Trixie gibt es noch nicht. Ein Eintrag ist aber vorbereitet und mit „#“ deaktiviert. Wenn die Backports live gehen, müsst ihr in der Datei nur die „#“ entfernen und ein **apt update** durchführen, um die Backports aktiv zu schalten.

Nach der Installation sieht eure **sources.list** so aus:

```
deb http://ftp.at.debian.org/debian/ trixie main non-free-firmware contrib non-free
#deb-src http://ftp.at.debian.org/debian/ trixie main non-free-firmware contrib non-free

deb http://security.debian.org/debian-security trixie-security main non-free-firmware contrib non-free
#deb-src http://security.debian.org/debian-security trixie-security main non-free-firmware contrib non-free

deb http://ftp.at.debian.org/debian/ trixie-updates main non-free-firmware contrib non-free
#deb-src http://ftp.at.debian.org/debian/ trixie-updates main non-free-firmware contrib non-free

# Backports (noch nicht verfügbar, später aktivieren)
# deb http://deb.debian.org/debian trixie-backports main contrib non-free non-free-firmware
```

Die Zeilen mit „*deb-src*“ habe ich mit „#“ deaktiviert, da sie in der Regel nicht benötigt werden.

Bei Fragen kannst du dich gerne an mich wenden: [techgamegeek@gmail.com](mailto:techgamegeek@gmail.com)

## Wayland deaktiviert

Wayland ist von Haus aus deaktiviert, da es vor allem in Verbindung mit NVIDIA Grafikkarten Probleme macht. Wenn ihr es aktivieren wollt, dann müsst ihr im Ordner `/usr/share/wayland-sessions/` die Datei **plasma.desktop.deactivated** auf **plasma.desktop** umbenennen.

## Plymouth deaktiviert

Leider macht auch „Plymouth“ in Verbindung mit einigen NVIDIA-Grafikkarten Probleme, weshalb es nicht installiert ist. Ihr seht deshalb beim Boot keinen schönen Splashscreen. Ich denke jedoch, das ist kein Problem – so bootet das System schneller. ;-)

## Nach dem Boot mit der LIVE-ISO nur „Grafikmatsch“?

Falls ihr beim Boot der Live-ISO (mit NVIDIA GPU) nur Grafikprobleme habt (kein Bild). Die Fenster nicht sieht, alles verschwommen ist, der Desktop nur aus Strichen besteht etc. Bitte versucht es mit der ...UEFI.iso die standardmäßig im „nomodeset“ Fallbackmodus bootet.