

**Erledigt**

## **Lenovo Thinkpad T430,T430s,X230,W530,T530 und die Akkus**

**Beitrag von „griven“ vom 16. Oktober 2018, 00:22**

Ich habe bei einem Hilfe Vor Ort Einsatz mein Thinkpad dabei gehabt und wie es der Teufel will kamen wir über die Thinkpads ins Gespräch und dem User bei dem ich zu Gast war fiel ein das er noch einen original verpackten Akku für ein T420 (6 Zellen/5200mAh) liegen hat den er aufgrund einer Fehlbestellung nicht verwenden kann (T420 passt nicht in T420s). Da die Akkus vom T420 und dem T430 physikalisch identisch sind sprich der T420 Akku ins T430 reinpasst hat mir der User den Akku mitgegeben. Zu Hause angekommen habe ich den Akku ins Notebook gepackt und siehe da die Akkus sind nicht nur physikalisch identisch sie versorgen das T430 auch artig mit Strom und der Akku lässt sich auch korrekt auslesen allerdings hat sich schnell Ernüchterung breit gemacht denn das T430 hat zwar gerne den Strom aus dem Akku genommen aber aufladen wollte es den Akku dann nicht mehr....

Soviel zur Vorgeschichte. Natürlich ist auch Lenovo nicht verborgen geblieben das die Stromspender aus dem T420 auch ins T430 passen oder eben die aus dem X220 auch ins X230 usw. und man sollte eigentlich meinen wie nachhaltig gedacht die Akkus passen in beide Generationen wenn man also aufrüstet kann man das teuer erworbene Zubehör weiter benutzen nur hat man in dem Fall die Rechnung ohne Lenovo gemacht. Lenovo hat natürlich gar kein Interesse daran besonders Nachhaltig zu agieren sondern das Interesse auf deren Seite liegt in einer Maximierung der Gewinne und damit wurde der Firmware des Embedded Controllers eine Whitelist verpasst die es dem Notebook verbietet die Akkus aus der Vorgängerserie zu laden. Die offizielle Rechtfertigung war natürlich das man ausschließen möchte das Nutzer potentiell unsichere Nachbauten verwenden kann man glauben muss man aber nicht...

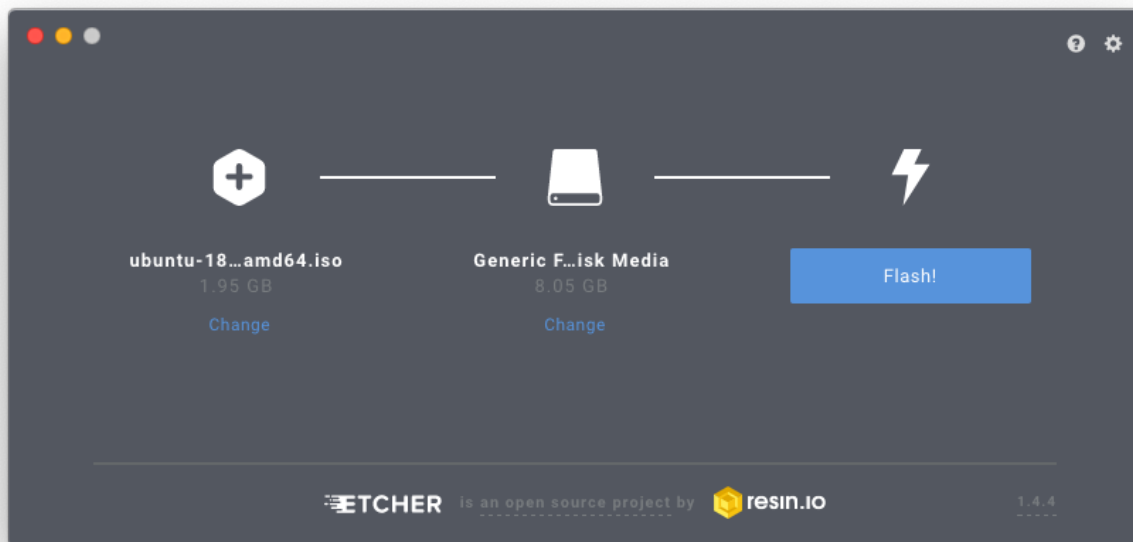
Wie wird man dieses Ärgernis jetzt los war dann die nächste Frage und die Antwort darauf lautet <https://github.com/hamishcoleman/thinkpad-ec> wobei es dabei in erster Linie darum geht ein klassisches Keyboard (7 Reihen) in ein Thinkpad mit den "neuen" Keyboards zu implantieren aber das muss uns an der Stelle nicht kümmern uns interessiert hierbei nur den ebenfalls in diesem Zug gefundenen Weg den Akkucheck in die Wüste zu schicken. Wie geht mal also nun vor, was wird gebraucht?

- 2 USB Sticks (min. 2GB für Linux und min. 512mb für den flasher)
- Das Programm Etcher <https://etcher.io>

- Ein Linux Live Image zum Beispiel Ubuntu <https://www.ubuntu.com/download/desktop>
- Bios mittels Windows zuvor auf die aktuellste Version bringen
- Im Bios den CSM Mode aktivieren und Boot Modus auf Legacy and UEFI einstellen

### **Schritt 1: Linux Stick erstellen:**

Ladet euch das Live Image herunter und während der Download läuft könnt Ihr schon mal einen USB Stick leer machen einfach den Stick im Festplattendienstprogramm als MBR und Fat32 formatieren der name ist egal. Ist der Download abgeschlossen und der Stick leer dann geht es mit dem hoffentlich schon geladenen Etcher weiter. Also Etcher starten und das eben heruntergeladene Image auswählen den USB Stick müsst Ihr nicht wählen wenn nicht mehr als einer am Rechner steckt denn dann erkennt Etcher diesen eigenständig als Ziel. hat man es richtig gemacht sieht es so aus:



den Vorgang nun einfach mit einem Klick auf Flash! starten und warten bis der Stick fertig erstellt ist. Je nachdem ob Ihr USB2 oder USB3 nutzt kann das einige Minuten dauern ist aber nicht schlimm denn während der Stick gebaut wird kann man ja hier schon mal weiterlesen 😊

### **Schritt 2: Linux auf dem Thinkpad booten:**

Wenn der Stick fertig ist wird macOS zum einen meckern das es mit dem Stick nichts anfangen kann und zum anderen sind wir jetzt bereit Linux auf dem Thinkpad zu starten also los geht es.

- Thinkpad abschalten falls noch nicht passiert

- Stick anschließen
- Thinkpad einschalten und F12 drücken um ins Bootmenu zu gelangen
- USB Stick auswählen
- Try Ubuntu without installing auswählen
- Nach einiger Zeit landen wir auf dem Desktop von Ubuntu

Haben wir es unfallfrei auf den Desktop geschafft geht es jetzt darum uns erstmal mit dem Internet zu verbinden. Sofern Ihr am Lan hängt sollte das automatisch passiert sein sofern ihr mit WLAN unterwegs seid muss eine Verbindung hergestellt werden was aber nur geht wenn Ihr keine von [Sascha 77](#) Rebrand Karten im Einsatz habt. Ich habe zu dem Zweck einfach einen USB WLAN Dongle mit dem ich ins Netz komme. Um Euch zu verbinden klickt oben rechts neben dem Batterie Indikator auf den nach unten gerichteten Pfeil und wählt dort Euer Wlan aus und verbindet Euch.

### **Schritt 3: Vorbereitungen treffen:**

Linux ist gestartet, das Internet verbunden dann wird es Zeit ein paar Vorbereitungen zu treffen denn wir werden mit dem Terminal arbeiten und dazu kann es nicht schaden wenn wir unter anderem eine deutsche Tastaturbelegung haben damit wir nicht ewig nach bestimmten Zeichen suchen müssen. Wir gehen wie folgt vor:

- Klick auf die 9 Punkte in der linken unteren Ecke
- Wechsel auf die 2 Seite
- Starten von Settings
- Auswahl von "Region & Language" und dann unter Input Sources German -> Germany hinzufügen
- Oben Rechts neben dem Lautsprecher Symbol von EN auf DE umstellen
- Fertig

Nun sind wir soweit das wir das Terminal öffnen können dazu wieder unten Links auf den 9 Punkte klicken und dann oben in der Mitte im Suchfeld Terminal eingeben und auf das gefundene Ergebnis klicken.

### **Schritt 4: nötige Pakete installieren und Quellen laden:**

Nun geht es darum nötige Pakete zu installieren und die Quellen aus dem Github Repo zu laden hierzu geben wir zunächst den folgenden Befehl ein um alle nötigen Pakete zu installieren:

Code

1. `sudo apt-get install build-essential git mtools libssl-dev`

und bestätigen die Frage ob wir die Pakete installieren wollen mit `y`. Wenn Die Pakete installiert sind geht es daran das die Quellen zu klonen auch hier verwenden wir wieder einen Terminal Befehl nämlich den folgenden:

Code

1. `git clone https://github.com/hamishcoleman/thinkpad-ec`

ist das erledigt geben wir abschließend noch

Code

1. `cd thinkpad-ec/`

ein damit wir auch in das Verzeichnis mit den so eben geladnen Quellen landen.

### **Schritt5: nötige Anpassungen vornehmen:**

Wie weiter oben schon geschrieben dient das Projekt ja eher dazu dem EmbeddedController das klassische Keyboard beizubringen und das Thema mit den Akkus ist eher zusätzlich mit dabei rausgekommen und so verwundert es eigentlich nicht weitet das just der Patch den wir wollen deaktiviert ist während alle die wir eigentlich nicht wollen aktiv sind. Wenn Ihr, wie in Schritt 4 erwähnt, in das Verzeichnis gewechselt habt dann könnt Ihr Euch jetzt mit dem Befehl `ls` gefolgt von Enter mal anschauen was da eigentlich so drinnen steckt. Neben einigen Dateien gibt es da auch diverse Verzeichnisse in die sich wiederum mit dem Befehl `cd Verzeichnisname/` wechseln könnt und in der Folge auch müsst. Ich gehe hier exemplarisch mal auf das T430 ein also das nackt ohne "s" oder sonstige Zusätze. Wie man sehen kann gibt es für das T430 zwei Verzeichnisse:

```
thinkpad-ec-master — -bash — 80x24
Last login: Mon Oct 15 19:15:24 on console
You have new mail.
[iMac:~ giesbertdumpich$ cd /Volumes/Daten/Downloads/thinkpad-ec-master ]
[iMac:thinkpad-ec-master giesbertdumpich$ ls ]
Descriptions.txt      docs                  t430s.G7HT39WW.img.d
LICENSE               mec-tools            t530.G4HT39WW.img.d
Makefile              radare               w530.G4HT39WW.img.d
README                scripts              x230.G2HT35WW.img.d
asm                   t430.G1HT34WW.img.d x230t.GCHT25WW.img.d
autoexec.bat.template t430.G1HT35WW.img.d
iMac:thinkpad-ec-master giesbertdumpich$
```

Beide repräsentieren unterschiedliche Bios Versionen und ich nicht genau weiß welche bei mir installiert ist und auch keine Lust habe nachzusehen bearbeite ich einfach beide Verzeichnisse gleichermaßen. Also was ist zu tun? Naja ganz einfach eigentlich zunächst wechselt man mit dem Befehl `cd /verzeichnisname` in das jeweilige Verzeichnis in meinem Beispiel ist das zunächst mit `cd t430.G1HT35WW.img.d/` für das erste Verzeichnis getan. Nach der Eingabe des Befehls `ls` ergibt sich folgendes Bild:

```
t430.G1HT35WW.img.d — -bash — 80x24
Last login: Mon Oct 15 19:15:24 on console
You have new mail.
[iMac:~ giesbertdumpich$ cd /Volumes/Daten/Downloads/thinkpad-ec-master ]
[iMac:thinkpad-ec-master giesbertdumpich$ ls ]
Descriptions.txt      docs                  t430s.G7HT39WW.img.d
LICENSE               mec-tools            t530.G4HT39WW.img.d
Makefile              radare               w530.G4HT39WW.img.d
README                scripts              x230.G2HT35WW.img.d
asm                   t430.G1HT34WW.img.d x230t.GCHT25WW.img.d
autoexec.bat.template t430.G1HT35WW.img.d
[iMac:thinkpad-ec-master giesbertdumpich$ cd t430.G1HT35WW.img.d/] ]
[iMac:t430.G1HT35WW.img.d giesbertdumpich$ ls ]
001_keysym.patch      004_fn_keys.patch
002_dead_keys.patch   005_fn_key_swap.patch
003_keysym_replacements.patch 006_battery_validate.patch.OFF
[iMac:t430.G1HT35WW.img.d giesbertdumpich$ ]
```

Wie man sehen kann gibt es hier diverse Dateien wobei auffällt das die Datei 006\_battery\_validate.patch.OFF sich von den anderen durch die Endung .OFF unterscheidet. Die Endung OFF sagt dem Script das dieser Patch nicht verwendet werden soll wobei die anderen schon benutzt werden sollen und genau das wollen wir ja nicht. Wir müssen jetzt also dafür sorgen das 006\_battery\_validate.patch.OFF verwendet wird der Rest aber nicht. Das erreichen wir indem wir erstmal alles aus dem Weg räumen was wir nicht haben wollen. Dazu gehen wir ein `rm *.patch` übrig bleibt 006\_battery\_validate.patch.OFF welchen es nun zu aktivieren gilt was wir erreichen indem wir eingeben

```
mv 006_battery_validate.patch.OFF 006_battery_validate.patch. Haben wir das in beiden Verzeichnissen gemacht können wir loslegen.
```

### **Schritt 5: EC Firmware patchen und Bootstick erstellen:**

Nun geht es ans Eingemachte wobei sich das schlimmer anhört als es ist. Wir wechseln zunächst mittels `cd .` wieder in das Verzeichnis thinkpad-ec und führen dort den Befehl `make clean` aus was folgende Ausgabe zur Folge hat:

```
thinkpad-ec-master — -bash — 80x24
[iMac:thinkpad-ec-master giesbertdumpich$ make clean
./scripts/generate_deps Descriptions.txt >.d/generated.deps
rm -f .d/generated.deps \
    patched.*.iso patched.*.img *.FL2 *.FL2.orig *.img.enc \
    *.img.enc.orig *.img.orig *.bat *.report \
    *.img \
    *.txt.orig
rm -rf *.iso.extract *.iso.orig.extract
iMac:thinkpad-ec-master giesbertdumpich$
```

ist das erledigt geht es gleich mit dem nächsten Befehl weiter wir geben ein `make patched.t430.img` und der Zauber beginnt und an seinem Ende steht ein fertiges Image für Euren Embedded Controller zumindest dann wenn die letzte Zeile mit Description: T430 Bios 2.77... anfängt oder eben passend zu eurem Modell. Bleibt eigentlich nur noch den Stick zu erstellen der Euch die FirmWare für den EC auf das Thinkpad bringt. Jetzt kommt der zweite USB Stick zum Einsatz den Ihr möglichst vorher gelöscht habt auch hier MBR und Fat und dem Ihr idealerweise den Namen flasher gegeben habt (der Name ist wichtig damit man den Stick auch findet). Um den Gerätenamen des Sticks zu ermitteln gebt Ihr nun folgenden Befehl ein `lsblk -d -o NAME,SIZE,LABEL` in meinem Fall hat der Stick den Gerätenamen `sdd` und den merke ich mir an der Stelle auch denn den brauch ich für den folgenden Befehl: `sudo dd if=patched.t430.img of=/dev/sdx` wobei Ihr das `sdx` durch Euren Gerätenamen ersetzen müsst in meinem Fall also durch `sdd` so das der Befehl in meinem Fall so aussieht: `sudo dd if=patched.t430.img of=/dev/sdd` Der Befehl schreibt das erstellte Image auf den Stick und wir sind fertig und können den Rechner ausschalten. Nachdem der Rechner ausgeschaltet wurde gehen wir wie folgt vor:

- Linux Stick abziehen
- **Netzteil anschließen**
- Rechner starten und mittels F12 das Bootmenu aufrufen
- Den vorher erstellten USB Stick auswählen

- Den Anweisungen folgen
- ggf. im Bios den CSM Mode wieder abschalten und UEFI only einstellen

Freuen das der Rechner von nun an mit den Akkus der älteren Generation spielt und zwar in beide Richtungen 😊