

## Begriff

Ein Dual-Boot-System ist ein Computer auf dem zwei Betriebssysteme installiert sind, mit einem Bootmanager der den Benutzer wählen lässt welches der beiden Systeme er gerade starten möchte. Eine typische Anwendung ist die parallele Installation von [Windows](#) und [Linux](#) auf einem PC. Wenn mehr als 2 Systeme installiert sind spricht man von Multi-Boot-Systemen.

## Dual-Boot bei der FRITZ!Box

Auch einige [FRITZ!Box-Modelle](#) sind Dual-Boot-Systeme, jedoch sind die Gründe ganz andere, und der Benutzer hat keinerlei Auswahlmöglichkeit zwischen den installierten Systemen.

Hintergrund ist die Tatsache dass bei einem [Update](#) der [Firmware](#) nichts schief gehen darf, sonst ist die Box in einem undefinierten Zustand und ein Fall für eine [Recovery](#). Dies passiert z.B. wenn man entgegen aller Warnungen während einem [Update](#) den Netzstecker zieht. Sofern ein [Update](#) manuell eingespielt wird führt dies nicht zwingend zu längeren Ausfällen, man bemerkt sehr schnell dass etwas schief lief und kann sich sofort helfen.

Bei automatischen [Updates](#) ist dies jedoch nicht so einfach, der Besitzer einer Box erfährt nicht wann ein solches eingespielt wird und ein Scheitern lässt sich normalerweise nicht automatisiert erkennen und führt zwangsweise zu Ausfällen bis der Besitzer den Schaden bemerkt. Die Lösung dieses Problems ist ein Dual-Boot-System bei dem ein [Update](#) das jeweils nicht aktive System überschreibt, und erst nach überprüfem Erfolg das neue System als aktiv markiert. Scheitert das [Update](#) aus irgendeinem Grund bleibt das aktive System unverändert und unbeschädigt. Solche Schreibvorgänge, die nie zu Datenverlust führen können, nennt man transaktionssicher.

Dieser Luxus hat natürlich seinen Preis, entweder man verbaut ein doppelt großes [Flash-EEPROM](#) oder man kürzt die [Firmware](#) auf die halbe Größe. In jedem Fall benötigt man zwei gleich große Partitionen für die beiden Systeme.

## Die Alice-IAD-Methode

Bei allen für [HanseNet/Alice](#) gefertigten FRITZ!Box-Derivaten wurde die [Firmware](#) gegenüber einer gleichartigen FRITZ!Box auf die Hälfte gekürzt. Die normalerweise für das System reservierte [Partition](#) in [mtd1](#) wurde in 2 Hälften zerteilt. Die jeweils aktive Hälfte ist [mtd1](#), die nicht aktive wird in [mtd5](#) gespeichert. Nach einem erfolgreichen [Update](#) werden die Partitionsdaten für [mtd1](#) und [mtd5](#) getauscht und ein [Neustart](#) ausgeführt, [ADAM2](#) startet dann wie immer die in [mtd1](#) eingetragene [Partition](#). Dieses Verfahren ist nahezu transaktionssicher, da das Tauschen der Partitionsdaten sequentiell abläuft (ideal wäre atomar) gibt es ein wenn auch sehr kurzes Zeitfenster in dem kein Strom ausfallen darf. Folgende Modelle nutzen diese Methode:

- [Alice IAD 5130](#)
- [Alice IAD WLAN 3331](#)
- [Alice IAD 7570](#)

Als Beispiel für bereits getauschte Partitionen siehe das [IAD 5130 Environment](#), und für noch nicht oder bereits doppelt getauschte Partitionen siehe das [IAD 7570 Environment](#).

## Die FRITZ!Box-Methode

Einige FRITZ!Box-Modelle verfügen über doppelt großes [Flash-EEPROM](#) oder nutzen ein geräumiges [NAND-Flash](#) zur Speicherung der [Firmware](#) und eignen sich daher hervorragend für ein Dual-Boot-System. Hier kommt eine andere Methode zum Einsatz. Im Gegensatz zur Alice-Methode wird im [Urlader-Environment](#) keine separate [Partition](#) für das zweite System definiert sondern in der Variable `linux_fs_start` mit '0' oder '1' angegeben welches System gestartet werden soll. Die Umschaltung übernimmt im [Kernel](#) der plattformspezifische [MTD-Treiber](#) bei der Erkennung und Zuweisung aller [Partitionen](#). Folgende Modelle nutzen diese Methode:

- [FRITZ!Box WLAN 3370](#)
- [FRITZ!Box 6360 Cable](#)
- [FRITZ!Box 6840 LTE](#)

## Die FRITZ!Fon-Methode

Auch alle neueren Handsets von [AVM](#) sind Dual-Boot-Systeme um transaktionssichere [Updates](#) zu gewährleisten. Da keine [Recovery](#) möglich ist würde ein gescheitertes [Update](#) das Gerät unbrauchbar machen. Die Handset-Firmware läuft direkt auf dem [DECT-SoC](#) ohne [Linux](#), [Partitionen](#) oder [MTD-Treiber](#). Das [Flash-EEPROM](#) steht der [CPU](#) des [SoC](#) als linear adressierbarer Speicher zur Verfügung. Darin werden dann zwei gleichgroße Bereiche für die Firmware reserviert. Die jeweils aktive wird beim Starten des Geräts über einen Vektor angesprungen. Bei einem [Update](#) wird dann der inaktive Bereich überschrieben. War dies erfolgreich wird der Startvektor umgebogen und das Gerät neugestartet.

Die Handset-Firmware ist kein relokatibler Code sondern für eine feste Basisadresse kompiliert, vermutlich läuft sie auf einem minimalistischen [RTOS](#) das über keine Relocation-Funktionen verfügt. Daher enthält ein Handset-Update immer 2 gleichgroße Binärdateien, ein High-Image und ein Low-Image. Diese sind funktionsidentische Kompilate für 2 verschiedene Basisadressen. Bei einem [Update](#) werden beide Kopien durch `start_dect_update.sh` entpackt. Es wird jedoch nur die Kopie für den beim Zielgerät gerade inaktiven Flash-Bereich benötigt. An welcher Stelle genau die Auswahl passiert wird noch erforscht, vermutlich erfährt `dect_manager` vom Handset welcher Bereich gerade aktiv ist. Folgende Modelle nutzen diese Methode:

- [FRITZ!Fon MT-D](#)
- [FRITZ!Fon MT-F](#)
- [FRITZ!Fon C3](#)
- [Speedphone 300](#)