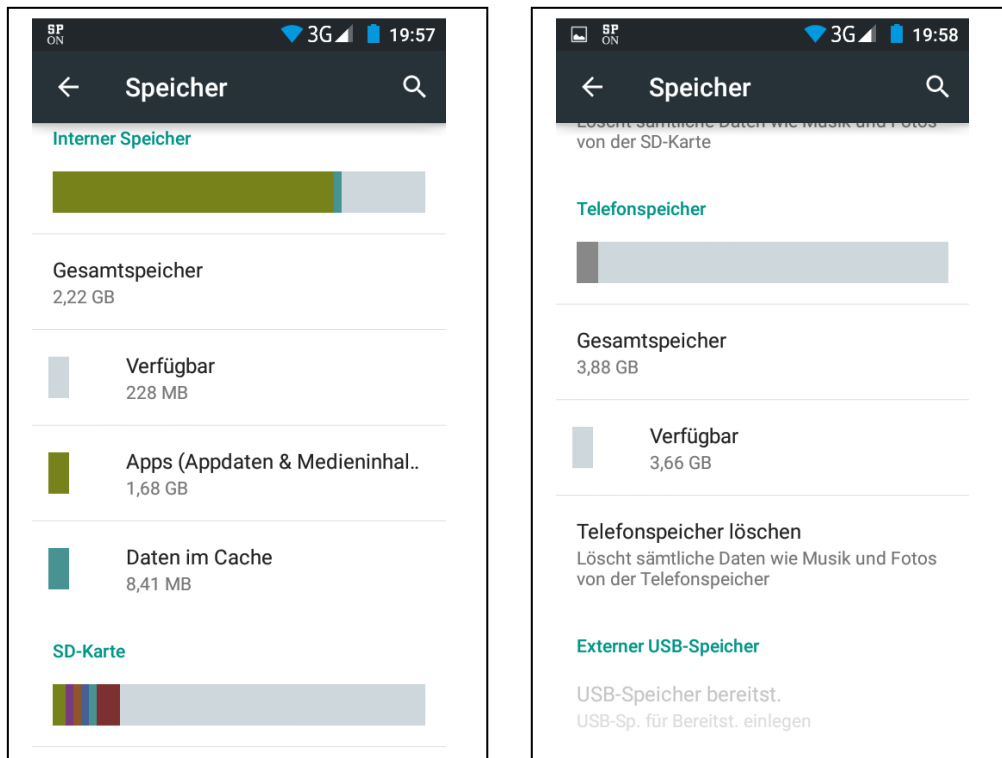


Vergrößern des internen Speichers durch Verkleinern des Telefonspeichers

Der interne Speicher des thl T6S ist mit 2GB knapp bemessen. Schon mit einer moderaten Anzahl von apps kommt man an Grenzen. Ärgerlich wird es, wenn Updates der apps anstehen, die aber wegen Speichermangels nicht durchgeführt werden können.

Verwendet man als Standardschreibdatenträger eine SD-Karte, bleibt der Telefonspeicher weitgehend ungenutzt. Es bietet sich daher an, den internen Speicher des Handys auf **3,5 GB bis 4 GB** zu vergrößern, indem man ihm Teile des Telefonspeichers zuschlägt.



Für die Realisierung erhielt ich auf der Plattform <https://www.android-hilfe.de> umgehend Unterstützung durch den *Super-Moderator & Mediator* **Wattsolls**. Vielen Dank dafür!

Das Thema wird unter <https://www.android-hilfe.de/thema/howto-mt65-xx-erweitern-internen-telefonspeicher-durch-ebr-flashing.606871/> behandelt.

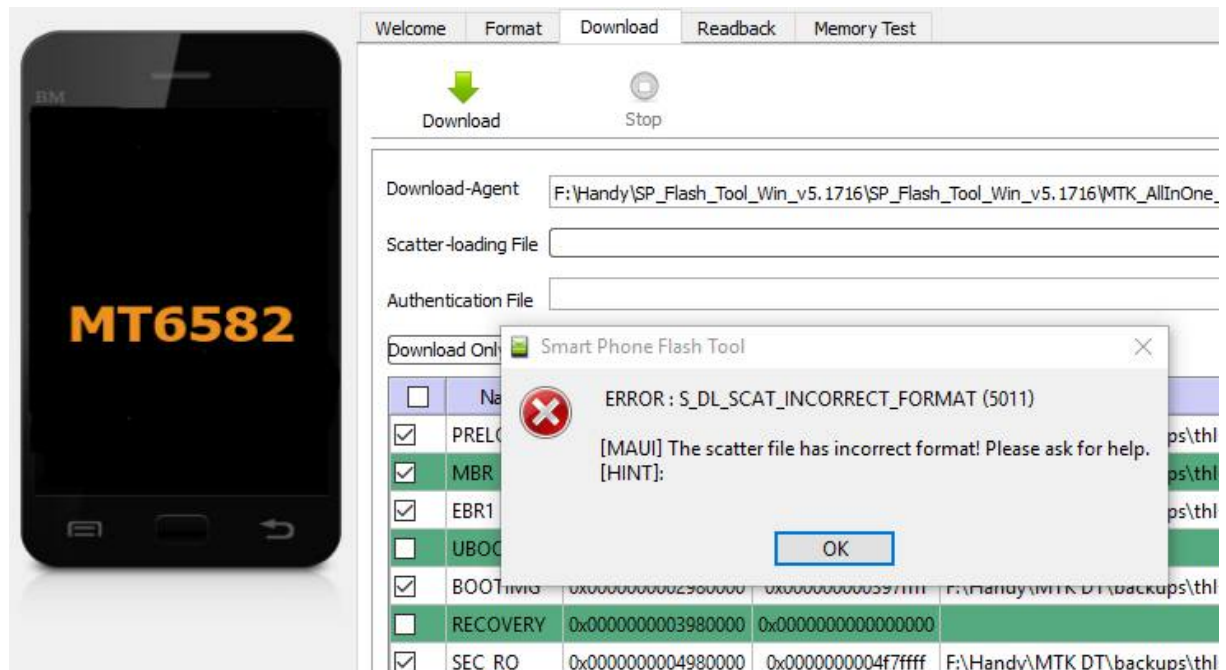
Benötigt werden:

1. Passendes ROM
2. SP Flashtool
3. VCOM Treiber
4. MTK Droidtools

Als ROM habe ich auf der Homepage von www.thlphone.com das aktuellste ROM für das thl T6S heruntergeladen:

(T6S-2016-2-25\thl.T6S.166K.1373MW.A3.151118.KK1.V2_10.FWVGA.COM.8P64.B1B5.rar).

Nach dem Entzippen und Einlesen der MT6582_Android_scatter.txt ergab sich ein Fehler wegen falschen Formats. Dieses ROM wurde daher verworfen.



Da die Auswahl eines geeigneten ROM unter NeedRom nicht gelang, wurde folgender Weg beschritten:

Sicherung des Stock ROM's

Ora beschreibt unter Tipp 5

<https://www.android-hilfe.de/thema/howto-mtk-droidtool-features-tipps-backup-ohne-root-root-imei-ect.535769/#post7173815>

Mit Hilfe der MTK Droidtool (DT) wird das Block Mapping bestimmt:

Block Info

Block devices Info								
N	Name	Scatter	Size	Byte Size	Type	Device	Mount Point	Mount Device
1	preloader	0000000000	0000C00000	12582912				
2	mbr	0000C00000	0000080000	524288		mmcblk0		
3	ebr1	0000C80000	0000080000	524288		mmcblk0p1		
4	pro_info	0000D00000	0000300000	3145728		mmcblk0		
5	nvrnm	0001000000	0000500000	5242880		mmcblk0		
6	protect_f	0001500000	0000A00000	10485760	ext4	mmcblk0p2	/protect_f	emmc@protect_f
7	protect_s	0001F00000	0000A00000	10485760	ext4	mmcblk0p3	/protect_s	emmc@protect_s
8	seccfg	0002900000	0000020000	131072		mmcblk0		
9	uboot	0002920000	0000060000	393216		mmcblk0		
10	booting	0002980000	0001000000	16777216		mmcblk0		
11	recovery	0003980000	0001000000	16777216		mmcblk0		
12	sec_ro	0004980000	0000600000	6291456		mmcblk0p4		
13	misc	0004F80000	0000080000	524288		mmcblk0		
14	logo	0005000000	0000300000	3145728		mmcblk0		
15	ebr2	0005300000	0000080000	524288		mmcblk0		
16	expdb	0005380000	0000A00000	10485760		mmcblk0		
17	android	0005D80000	0038400000	943718400	ext4	mmcblk0p5	/system	emmc@android
18	cache	003E180000	0007E00000	132120576	ext4	mmcblk0p6	/cache	emmc@cache
19	usrdata	0045F80000	0092E00000	2464153600	ext4	mmcblk0p7	/data	emmc@usrdata
20	fat	00D8D80000	00F8980000	4170711040		mmcblk0p8		
21	bmtpool	00FFFF00A8	0001500000	22020096		mmcblk0		

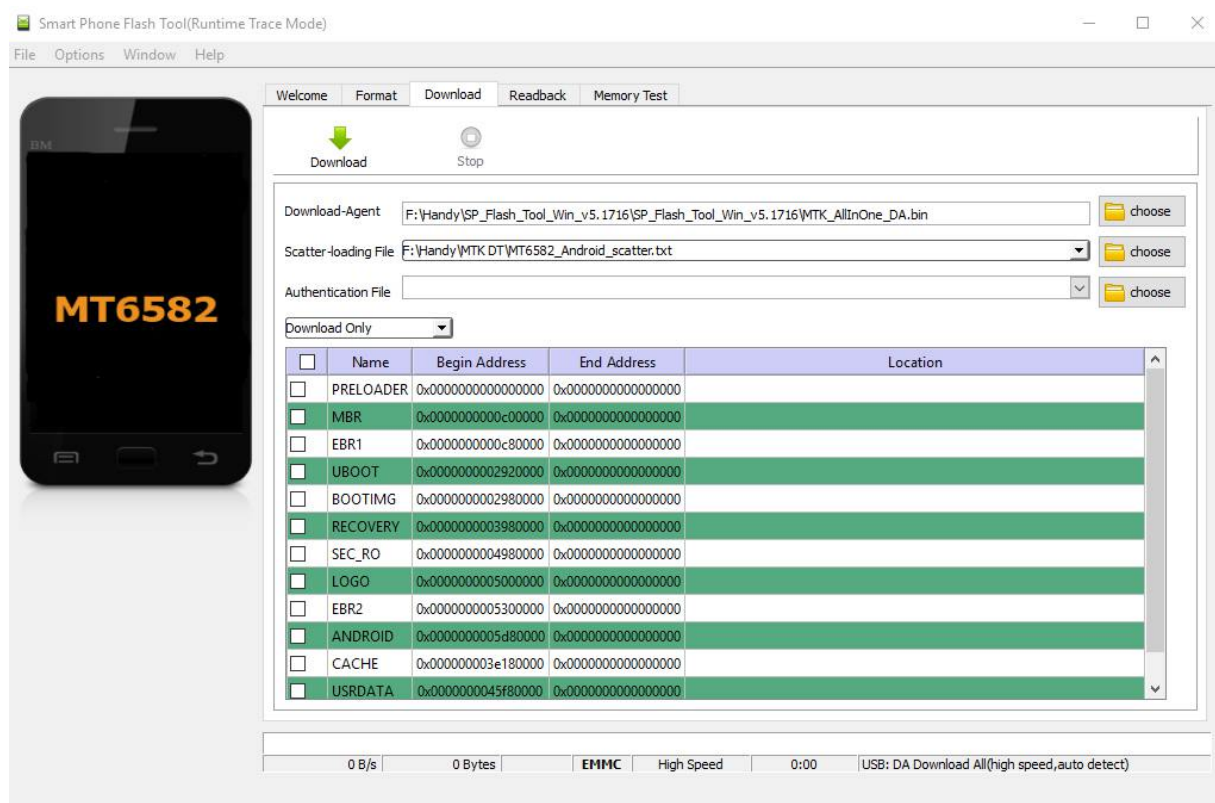
Mit einem Click auf „Scatter aus Datei“ wird die Datei „MT6582_Android_scatter.txt“ erzeugt wird.

```
- partition_index: SYS17
partition_name: CACHE
file_name: cache.img
is_download: true
type: VAFES_IMG
linear_start_addr: 0x3E180000
physical_start_addr: 0x30580000
partition_size: 0x7E00000
region: EMMC_USER
storage: HW_STORAGE_EMMC
boundary_check: true
is_reserved: false
operation_type: UPDATE
reserve: 0x00
```

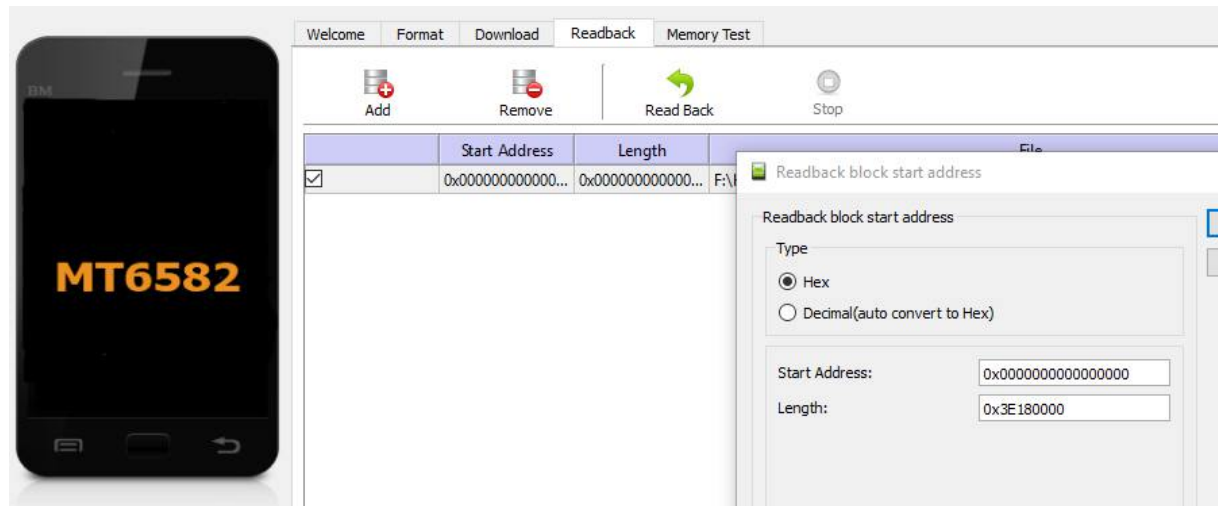
Unter „CACHE“ wird die „linear_start_addr“ in die Zwischenablage kopiert.

Sicherung des ROM's

Mit SP Flashtool die gerade erzeugte scatter.txt auswählen. Dabei darauf achten, dass alle Auswahlboxen deaktiviert sind.



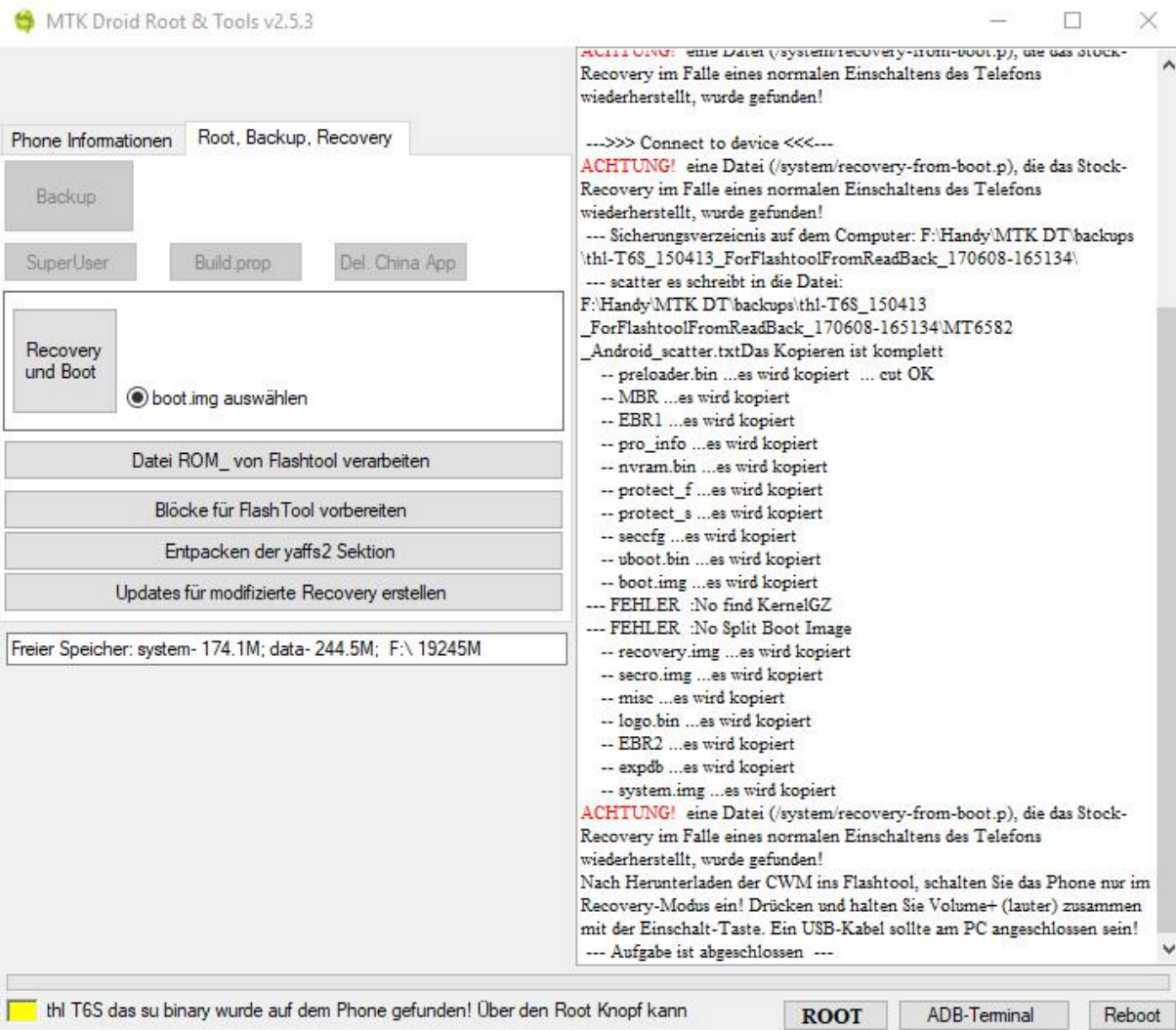
Auf den Reiter Read Back navigieren, alle Einträge mit remove entfernen und einen neuen Eintrag anlegen. Name für das ROM eingeben und mit „ReadBack“ starten. USB koppeln.



Nachbereiten der Sicherung mittels MTK Droidtool (Segmentieren für Flashtool)

1. Unter dem zweiten Tab des MTK Droid den Button "Datei ROM_ vom Flashtool verarbeiten" drücken.
2. Die Segmentierung, das Erzeugen einer Scatterfile beginnt. Nach einer Weile wird gefragt, ob das CWM-R Recovery "automatisch" erzeugt werden soll.
3. Leider fehlt der Link, um eine aktuellere Vorlage, als die im MTK Droidtool verwendete, zu nutzen.
4. Also empfehle ich hier gleich eine andere Vorlage, also eine dem Prozessor adäquate neuere Version des recovery.img, zu wählen. Beispiele findet in im CWM Recovery Threat (siehe meine Signatur). Leider nicht gefunden

Sind die Meldungen im u.s. Bild i.O. oder sind die Fehlermeldungen kritisch?



Als Ergebnis der „automatisch“ erzeugten CWM-R Recovery sind diese Files entstanden:

EBR1	512
EBR2	512
expdb	10 485 760
MBR	512
misc	524 288
pro_info	3 145 728
protect_f	10 485 760
protect_s	10 485 760
seccfg	131 072
logo.bin	3 145 728
nvram.bin	5 242 880
preloader.bin	119 248
uboot.bin	393 216
boot.img	16 777 216
factory_NONmodified_recovery.img	16 777 216
secro.img	6 291 456
system.img	943 718 400
MT6582_Android_scatter.txt	7 624

Sind die links stehenden Files vollständig für einen Flash?

Es müssen wohl noch EBR1, EBR2 und scatter.txt angepasst werden