

Samsung Galaxy S5 (SM-G900F)

Das Smartphone „**Samsung Galaxy S5 (SM-G900F)**“ lässt sich folgendermaßen auf LineageOS umstellen:

- Installation des Recovery-Abbildes [TWRP](#)
- Administratorrechte mit [Magisk](#)
- Installation von [LineageOS](#)

Wichtige Links hierzu:

- Stock-ROM:
 - [android-digital.de](#)
 - [samsung-updates.com](#)
 - [android-hilfe.de](#) (Melonos)
 - [android-hilfe.de](#) (submain)
 - [androidfilehost.com](#)
 - [usp-forum.de](#)
 - [updato.com](#)
 - [xda-developers.com](#)
 - [sammobile.com](#)
 - [samsung-firmware.org](#)
 - [handy-faq.de](#)
 - [pocketpc.ch](#)
 - [go41.de](#)
 - [android-hubo.de](#)
 - [faq4mobiles.de](#)
- Bauen unter DEBIAN
 - [github.com](#)
- Custom ROM
 - [LineageOS_15.1](#)

Recovery-Abbild (TWRP)

Ist das Smartphone noch nicht mit Administrationsrechten ausgestattet („rooted“), gibt es die Möglichkeit das TWRP über Odin zu installieren. Unter Linux wird dafür das Programm [Heimdall](#) verwendet, welches auf ähnliche Weise arbeitet.

Download-Modus

Damit das Gerät erkannt wird, muss dieses in den sogenannten „Download-Modus“ gebracht werden:

- Gerät ausschalten
- gleichzeitiges Drücken und Halten der Lautstärke-Leiser-Taste („volume down“) (linke Seite, Taste nach unten) + der Home-Taste („home“) (mittlere große Taste unten) + der Anschalt-Taste („power“) (rechte Seite)
- es erscheint eine Warnmeldung, die aber durch Drücken von Lautstärke-Lauter übersprungen werden kann

TWRP herunterladen

Jetzt kann das TWRP heruntergeladen werden, passend zum Gerät: [twrp-3.2.1-0-klte.img](#). Dieses wird dann später über Heimdall auf das Gerät gebracht und installiert.

PIT-Datei erstellen

Weiterhin benötigt Heimdall eine sogenannte PIT-Datei (PIT: Partition Information Table), die hier in dem Fall vom Gerät selbst heruntergeladen wurde. Dafür kann das grafische Front-End oder das Konsolenprogramm verwendet werden:

```
~# heimdall download-pit --output samsung-galaxy-alpha-sm-g900f.pit
Heimdall v1.4.2
...
Downloading device's PIT file...
WARNING: Empty bulk transfer after receiving packet failed. Continuing
anyway...
PIT file download successful.

Ending session...
Rebooting device...
Releasing device interface...
```

TWRP installieren

Jetzt kann das TWRP auf das Gerät installiert werden. Die grafische Methode sieht wie folgt aus:

- Heimdall-Programm (Frontend) als Administrator starten
- unter dem Reiter „Utilities“ auf „Detect“ drücken, damit das Gerät erkannt wird
 - ist das erfolgt, erscheint im „Output“: „Device detected“
- unter dem Reiter „Flash“ bei „PIT“ über „Browse“ die eben erstellte PIT-Datei laden
- dann rechts unter „Partitions (Files)“ auf „Add“ klicken und links bei „Partition Name“ den Punkt „RECOVERY“ auswählen

- jetzt unter „File (recovery.img)“ die heruntergeladene TWRP-Abbild laden
- bevor die Installation durchgeführt wird, sollte noch der Haken bei „No Reboot“ gesetzt werden, damit das Telefon nicht automatisch neu startet
- die Installation wird mit dem Drücken von „Start“ (unten rechts) durchgeführt
 - der Erfolg der Installation wird mit „Flash completed successfully!“ angezeigt und das Telefon startet neu

Die Installation kann ebenfalls per Konsole durchgeführt werden:

```
~# heimdall flash --RECOVERY twrp-3.2.1-0-klte.img
```

Nach der erfolgreichen Installation ist es ganz wichtig, dass das Telefon sofort in das Recovery-Abbild bootet, sonst erfolgt ein Überschreiben des Abbildes. So kann das Telefon per Tastendruck in den Recovery-Modus gebracht werden:

- gleichzeitiges Drücken und Halten der Lautstärke-Lauter-Taste („volume up“) (linke Seite, Taste nach oben) + der Home-Taste („home“) (mittlere große Taste unten) + der Anschalt-Taste („power“) (rechte Seite)

Administratorrechte mit Magisk

Dieser Schritt ist für die Installation eines LineageOS-Abbildes nicht unbedingt notwendig. Soll das Gerät trotzdem ge-„rooted“ werden, kann das mit dem Programm „Magisk“ durchgeführt werden. Die aktuelle Version findet sich auf der Webseite gleich im ersten [Link](#). Die ZIP-Datei wird auf das Gerät übertragen (oder über das Telefon direkt heruntergeladen) und im Wurzelverzeichnis des internen Speichers abgelegt.

Jetzt muss das Gerät in den Recovery-Modus gebracht werden:

- Gerät ausschalten
- gleichzeitiges Drücken und Halten der Lautstärke-Lauter-Taste („volume up“) (linke Seite, Taste nach oben) + der Home-Taste („home“) (mittlere große Taste unten) + der Anschalt-Taste („power“) (rechte Seite)

Im TWRP kann dann unter „Install“ die heruntergeladene ZIP-Datei ausgewählt werden und mit „Swipe to Confirm Flash“ die Installation gestartet werden. Nach erfolgreicher Installation kann das Gerät neu gestartet werden.

LineageOS

Das Abbild von LineageOS kann [hier](#) heruntergeladen werden. Die aktuelle Version ist [lineage-14.1-20180427-nightly-klte-signed.zip](#).

Die heruntergeladene ZIP-Datei muss auf das Telefon gebracht werden, wieder ins Wurzelverzeichnis des internen Speichers. Soll dies im Recovery-Modus geschehen, kann dafür „adb“ verwendet werden:

```
~# adb push lineage-14.1-20180427-nightly-klte-signed.zip /sdcard/  
[100%] /sdcard/lineage-14.1-20180427-nightly-klte-signed.zip
```

Wurde der Recovery-Modus bereits wieder beendet, muss es erneut in diesen Modus gebracht werden:

- Gerät ausschalten
- gleichzeitiges Drücken und Halten der Lautstärke-Lauter-Taste („volume up“) (linke Seite, Taste nach oben) + der Home-Taste („home“) (mittlere große Taste unten) + der Anschalt-Taste („power“) (rechte Seite)

Dort werden dann folgende Schritte durchgeführt:

- im Hauptbildschirm auf „Löschen“ und den Schalter unten nach rechts schieben um die Werkseinstellungen herzustellen
- dann auf „Zurück“ und über die Rücktaste (Pfeil nach links) um wieder auf den Hauptbildschirm zu gelangen
- dann auf „Installieren“ gehen und das heruntergeladene Abbild („lineage-14.1-20180427-nightly-klte-signed.zip“) auswählen
- den Schalter unten nach rechts schieben, um die Installation zu starten
- jetzt kann das Telefon über „System neustarten“ neu gestartet werden

Stock-ROM

Sollte der Fall auftreten, dass es notwendig wird, die originale Software (das sogenannte Stock-ROM) wieder auf das Telefon zu bringen, ist dies ebenfalls über Heimdall möglich. Dafür muss das Stock-ROM zuerst heruntergeladen werden. Es gibt dafür im Internet mehrere, teils nicht so vertrauenswürdige, Quellen. Nachfolgend wurde das ROM von [samsung-firmware.org](#) heruntergeladen: [DBT-G900FXXS1CQD4-20170505133038.zip](#). Das ZIP-Archiv wird nach dem Download entpackt:

```
~# unzip DBT-G900FXXS1CQD4-20170505133038.zip  
Archive:  DBT-G900FXXS1CQD4-20170505133038.zip
```

```
inflating: G900FXXS1CQD4_G900FDBT1CQA1_G900FXXU1CQA3_HOME.tar.md5
```

Das entpackte TAR-Archiv muss ebenfalls entpackt werden:

```
~# tar xvf G900FXXS1CQD4_G900FDBT1CQA1_G900FXXU1CQA3_HOME.tar.md5
aboot.mbn
NON-HLOS.bin
rpm.mbn
sbl1.mbn
tz.mbn
boot.img
recovery.img
system.img.ext4
modem.bin
cache.img.ext4
hidden.img.ext4
```

Das Telefon muss dann wieder in den Download-Modus gebracht werden:

- Gerät ausschalten
- gleichzeitiges Drücken und Halten der Lautstärke-Leiser-Taste („volume down“) (linke Seite, Taste nach unten) + der Home-Taste („home“) (mittlere große Taste unten) + der Anschalt-Taste („power“) (rechte Seite)
- es erscheint eine Warnmeldung, die aber durch Drücken von Lautstärke-Lauter übersprungen werden kann

Damit die einzelnen Abbilder den richtigen Partitionen im Telefon zugewiesen werden können, kann die Partionstabelle angezeigt werden:

```
~# heimdall print-pit
Heimdall v1.4.2

...

Session begun.

Downloading device's PIT file...
PIT file download successful.

Entry Count: 30
Unknown 1: 1598902083
Unknown 2: 844251476
Unknown 3: 21325
Unknown 4: 14413
Unknown 5: 14137
Unknown 6: 52
```

```
Unknown 7: 0
Unknown 8: 0

...

--- Entry #29 ---
Binary Type: 0 (AP)
Device Type: 2 (MMC)
Identifier: 73
Attributes: 5 (Read/Write)
Update Attributes: 1 (FOTA)
Partition Block Size/Offset: 30777311
Partition Block Count: 33
File Offset (Obsolete): 0
File Size (Obsolete): 0
Partition Name: SGPT
Flash Filename: sgpt.img
FOTA Filename:

Ending session...
Releasing device interface...
```

Daraus ergeben sich für das Gerät folgende wichtigen Einträge:

- Eintrag #0:
 - Partition Name: APNHL0S
 - Flash Filename: NON-HLOS.bin
- Eintrag #1:
 - Partition Name: MODEM
 - Flash Filename: modem.bin
- Eintrag #2:
 - Partition Name: SBL1
 - Flash Filename: sbl1.mbn
- Eintrag #5:
 - Partition Name: AB00T
 - Flash Filename: aboot.mbn
- Eintrag #6:
 - Partition Name: RPM
 - Flash Filename: rpm.mbn
- Eintrag #7:
 - Partition Name: TZ
 - Flash Filename: tz.mbn
- Eintrag #14:
 - Partition Name: B00T
 - Flash Filename: boot.img
- Eintrag #15:
 - Partition Name: RECOVERY
 - Flash Filename: recovery.img

- Eintrag #22:
 - Partition Name: SYSTEM
 - Flash Filename: system.img.ext4
- Eintrag #23:
 - Partition Name: CACHE
 - Flash Filename: cache.img.ext4
- Eintrag #24:
 - Partition Name: HIDDEN
 - Flash Filename: hidden.img.ext4

Bevor das Installieren der Abbilder beginnt, wird geprüft, ob das Telefon von Heimdall korrekt erkannt wird:

```
~# heimdall detect
Device detected
```

Jetzt kann mittels des Programms „heimdall“ das Gerät mit den entpackten Dateien beschrieben werden. Die Kommandozeile ergibt sich aus den eben ermittelten Einträgen:

```
~# heimdall flash --APNHL0S NON-HLOS.bin --MODEM modem.bin --SBL1 sbl1.mbn -
-ABOOT aboot.mbn \
--RPM rpm.mbn --TZ tz.mbn --BOOT boot.img --RECOVERY recovery.img --SYSTEM
system.img.ext4 --CACHE cache.img.ext4 \
--HIDDEN hidden.img.ext4
```

Das Telefon startet nach der Installation automatisch neu.

LineageOS 14.1

Nachfolgend soll das Erstellen eines selbstgebauten Android-Abbildes (Custom ROM) für das Smartphone „Samsung Galaxy S5 (SM-G900F)“ beschrieben werden. Dafür wird das GIT-Repository der Seite <https://github.com/LineageOS/android.git> mit dem Branch „cm-14.1“ genutzt.

Nützliche Webseiten

- lineageos.org

Vorbereitung

Der interne Codename des Gerätes muss ermittelt werden, was auf der [LineageOS](#)-Seite herausgefunden werden kann. Das oben genannte Gerät hat den Codenamen „klte“.

Zur weiteren Vorbereitung gehört das Erstellen der Verzeichnisstruktur und die Synchronisation des Repository, wie [hier](#) beschrieben.

Quellcode herunterladen

Im nächsten Schritt erfolgt das Einlesen der Bauumgebung:

```
~$ cd ~/AndroidCustomROM/cm-14.1
~$ source build/envsetup.sh
including device/generic/mini-emulator-arm64/vendorsetup.sh
including device/generic/mini-emulator-armv7-a-neon/vendorsetup.sh
including device/generic/mini-emulator-x86/vendorsetup.sh
including device/generic/mini-emulator-x86_64/vendorsetup.sh
including vendor/cm/vendorsetup.sh
including sdk/bash_completion/adb.bash
including vendor/cm/bash_completion/git.bash
including vendor/cm/bash_completion/repo.bash
```

Jetzt kann der gerätespezifische Quellcode heruntergeladen werden:

```
~$ breakfast klte
including vendor/cm/vendorsetup.sh
build/core/product_config.mk:249: *** Can not locate config makefile for
product "lineage_klte". Stop.
build/core/product_config.mk:249: *** Can not locate config makefile for
product "cm_klte". Stop.
build/core/product_config.mk:249: *** Can not locate config makefile for
product "lineage_klte". Stop.
Device klte not found. Attempting to retrieve device repository from
LineageOS Github (http://github.com/LineageOS).
Found repository: android_device_samsung_klte
Default revision: cm-14.1
Checking branch info
...
HOST_CROSS_2ND_ARCH=x86_64
HOST_BUILD_TYPE=release
```



```
BUILD_ID=NJH47F
OUT_DIR=~/.AndroidCustomROM/cm-14.1/system/out
=====
```

Gerätespezifikationen

Es wird ein LineageOS-Abbild für das Gerät benötigt, welches zum Beispiel von der Seite <https://download.lineageos.org/klte> heruntergeladen werden kann:

```
~$ wget
https://mirrorbits.lineageos.org/full/klte/20180427/lineage-14.1-20180427-nightly-klte-signed.zip
```

Die notwendige Verzeichnisstruktur erstellen:

```
~$ mkdir ~/.AndroidCustomROM/system_dump
~$ cd ~/.AndroidCustomROM/system_dump
```

Die benötigten Dateien extrahieren:

```
~$ unzip lineage-14.1-20180427-nightly-klte-signed.zip system.transfer.list
system.new.dat
```

Jetzt kann das System-Image umgewandelt werden:

```
~$ python ~/.AndroidCustomROM/bin/sdat2img.py system.transfer.list
system.new.dat system.img
```

Dann ein Verzeichnis anlegen und einhängen:

```
~$ mkdir system/
~$ sudo mount system.img system/
```

Jetzt wieder zur Wurzel zurückspringen und die Dateien entpacken:

```
~$ cd ../cm-14.1/device/samsung/klte
~$ ./extract-files.sh ~/.AndroidCustomROM/system_dump/
```

Das eingehängte Image kann wieder ausgehängt und das erstellte Verzeichnis entfernt werden:

```
~$ sudo umount ~/AndroidCustomROM/system_dump/system
~$ rm -rf ~/AndroidCustomROM/system_dump
```

Image bauen

Jetzt kann das Image bebaut werden:

```
~# croot
~# brunch klte
including vendor/cm/vendorsetup.sh
Looking for dependencies in device/samsung/klte
Looking for dependencies in device/samsung/klte-common
Looking for dependencies in device/samsung/msm8974-common
Looking for dependencies in device/qcom/common
Dependencies file not found, bailing out.
Looking for dependencies in device/samsung/qcom-common

=====
PLATFORM_VERSION_CODENAME=REL
PLATFORM_VERSION=7.1.2
LINEAGE_VERSION=14.1-20180227-UNOFFICIAL-klte
...
using prebuilt boot.img from B00TABLE_IMAGES...
  boot size (8925200) is 65.47% of limit (13631488)
  running: openssl pkcs8 -in build/target/product/security/testkey.pk8 -
inform DER -nocrypt
  running: java -Xmx2048m -
Djava.library.path=~/.AndroidCustomROM/cm-14.1/system/out/host/linux-
x86/lib64 -jar \
  ~/.AndroidCustomROM/cm-14.1/system/out/host/linux-x86/framework/signapk.jar
-w build/target/product/security/testkey.x509.pem \
  build/target/product/security/testkey.pk8 /tmp/tmpXmmEKi \
  ~/.AndroidCustomROM/cm-14.1/system/out/target/product/klte/lineage_klte-
ota-6153c7d683.zip
done.
[100% 34558/34558] build bacon
Package Complete:
~/.AndroidCustomROM/cm-14.1/system/out/target/product/klte/lineage-14.1-20180
227-UNOFFICIAL-klte.zip
make: Leaving directory '~/.AndroidCustomROM/cm-14.1/system'

#### make completed successfully (47:46 (mm:ss)) ####
```

Ergebnis

Verlief das Bauen erfolgreich, befinden sich im Verzeichnis „out/target/product/klte/“ (entspricht der Variablen „\$OUT“) unter anderem die zwei wichtigen Dateien:

- „recovery.img“: Das LineageOS-Recovery-Image
- „lineage-14.1-20180227-UNOFFICIAL-klte.zip“: Das LineageOS-Installationsimage

Resurrection Remix OS (Nougat)

Nachfolgend soll das Erstellen eines selbstgebauten Android-Abbildes (Custom ROM) für das Smartphone „Samsung Galaxy S5 (SM-G900F)“ beschrieben werden. Dafür wird das GIT-Repository der Seite https://github.com/ResurrectionRemix/platform_manifest.git mit dem Branch „nougat“ verwendet. Der [ResurrectionRemix](#) ist eine Alternative zu LineageOS.

Nützliche Webseiten

- [ResurrectionRemixOS](#)
- [GitHub](#)
- [YouTube](#)
- [xda-developers.com](#)

Vorbereitung

Zur weiteren Vorbereitung gehört das Erstellen der Verzeichnisstruktur und die Synchronisation des Repository, wie [hier](#) beschrieben.

Als Arbeitsverzeichnis wird das Verzeichnis „~/AndroidCustomROM/rr-nougat“ erstellt und verwendet.

Die Initialisierung des Repository sieht dann so aus:

```
~$ repo init -u https://github.com/ResurrectionRemix/platform_manifest.git -
```

b nougat

Gerätespezifikationen

Die geräte-spezifischen Daten können auf der GIT-Webseite heruntergeladen werden. Das gleiche gilt für die Kernel- und Vendor-Daten.

Gerätedaten

Herunterladen der geräte-spezifischen Daten von [GitHub](#) durch Eingabe des Gerätes im Suchfeld (hier in dem Fall „[samsung klte](#)“:

```
~$ git clone https://github.com/LineageOS/android_device_samsung_klte.git -b  
cm-14.1 ~/AndroidCustomROM/rr-nougat/device/samsung/klte  
Cloning into '~/AndroidCustomROM/rr-nougat/device/samsung/klte'...  
remote: Counting objects: 1254, done.  
remote: Total 1254 (delta 0), reused 0 (delta 0), pack-reused 1253  
Receiving objects: 100% (1254/1254), 769.79 KiB | 319.00 KiB/s, done.  
Resolving deltas: 100% (653/653), done.
```

Kerneldaten

Herunterladen der Kerneldaten von [GitHib](#) durch Eingabe des Kernelnames im Suchfeld (hier in dem Fall „[samsung msm8974](#)“:

```
~$ git clone https://github.com/LineageOS/android_kernel_samsung_msm8974 -b  
cm-14.1 ~/AndroidCustomROM/rr-nougat/kernel/samsung/msm8974  
Cloning into '~/AndroidCustomROM/rr-nougat/kernel/samsung/msm8974'...  
remote: Counting objects: 2800088, done.  
remote: Compressing objects: 100% (18/18), done.  
remote: Total 2800088 (delta 8), reused 6 (delta 3), pack-reused 2800067  
Receiving objects: 100% (2800088/2800088), 588.53 MiB | 770.00 KiB/s, done.  
Resolving deltas: 100% (2327440/2327440), done.  
Checking out files: 100% (45831/45831), done.
```

Vendordaten

Herunterladen vom Vendor von [GitHub](#) durch Eingabe des Geräteherstellers im Suchfeld (hier in dem Fall „samsung“:

```
~$ git clone https://github.com/TheMuppets/proprietary_vendor_samsung -b
cm-14.1 ~/AndroidCustomROM/rr-nougat/vendor/samsung
Cloning into '~/AndroidCustomROM/rr-nougat/vendor/samsung'...
remote: Counting objects: 38323, done.
remote: Compressing objects: 100% (13/13), done.
remote: Total 38323 (delta 2), reused 4 (delta 2), pack-reused 38308
Receiving objects: 100% (38323/38323), 1.92 GiB | 701.00 KiB/s, done.
Resolving deltas: 100% (19451/19451), done.
Checking out files: 100% (10084/10084), done.
```

Abbild bauen

Jetzt kann die Bauumgebung initialisiert werden:

```
~$ cd ~/AndroidCustomROM/rr-nougat
~# source build/envsetup.sh
including device/generic/mini-emulator-arm64/vendorsetup.sh
including device/generic/mini-emulator-armv7-a-neon/vendorsetup.sh
including device/generic/mini-emulator-x86_64/vendorsetup.sh
including device/generic/mini-emulator-x86/vendorsetup.sh
including vendor/cm/vendorsetup.sh
including sdk/bash_completion/adb.bash
including vendor/cm/bash_completion/git.bash
including vendor/cm/bash_completion/repo.bash

▼ Which build-type do you want to choose?
☐ 1- Official      ☐ Only for the official builds by the maintainers
☐ 2- Unofficial   ☐ For the unofficial builds
☐ 3- Nightly     ☐ Only for the official daily builds by the maintainers
☐ 4- Weekly      ☐ Only for the official weekly builds by the maintainers
☐ 5- Release     ☐ Only for the official releases by the RR-Team
☐ 6- Milestone   ☐ Only for the milestone releases by the RR-Team
☐ 7- Experimental ☐ Only for the experimental releases by the RR-Team
☐ 8- Final       ☐ Only for the final releases by the RR-Team
☐ 30/sec Time-out ☐ Default option

▼ Pick a number
2

■ Building unofficial variant
```

```

▼ Which root method do you want to use?
☐ 0- Magisk ☐ Magisk Root Method
☐ 1 or any other input- Rootless ☐ Unrooted builds [default]
☐ 30/sec Time-out ☐ Default option
▼ Pick a number
1

■ Rootless build

▼ For how many days changelog do you want to generate?
☐ 30/sec Time-out ☐ 7 Days by default

▼ Type a number
1

☐ Generating day number 1 ■ 04/29/2018..
✓ Changelog successfully generated.

```

Jetzt kann das Abbild ausgewählt werden:

```

~# lunch
You're building on Linux

Lunch menu... pick a combo:
 1. aosp_arm64-eng
87. lineage_kltesprsprts-userdebug
 2. aosp_arm-eng
88. lineage_klte-userdebug
 3. aosp_mips64-eng
89. lineage_kuntao-userdebug
 4. aosp_mips-eng
90. lineage_land-userdebug
 5. aosp_x86_64-eng
91. lineage_lentislte-userdebug
 6. aosp_x86-eng
92. lineage_lettuce-userdebug
 7. lineage_a5y17lte-userdebug
93. lineage_libra-userdebug
 8. lineage_a7y17lte-userdebug
94. lineage_lithium-userdebug
 9. lineage_addison-userdebug
95. lineage_ls990-userdebug
10. lineage_armani-userdebug
96. lineage_ls997-userdebug
11. lineage_athene-userdebug
97. lineage_lt02ltespr-userdebug
12. lineage_axon7-userdebug
98. lineage_lt03lte-userdebug
44. lineage_gts210wifi-userdebug
130. lineage_seed-userdebug
45. lineage_h830-userdebug
131. lineage_serrano3gxx-userdebug
46. lineage_h850-userdebug
132. lineage_serranodsdd-userdebug
47. lineage_h870-userdebug
133. lineage_serranoltexx-userdebug
48. lineage_h910-userdebug
134. lineage_shieldtablet-userdebug
49. lineage_h918-userdebug
135. lineage_spyder-userdebug
50. lineage_ha3g-userdebug
136. lineage_sumire-userdebug
51. lineage_hammerhead-userdebug
137. lineage_surnia-userdebug
52. lineage_ham-userdebug
138. lineage_suzuran-userdebug
53. lineage_harpia-userdebug
139. lineage_t0lte-userdebug
54. lineage_hayabusa-userdebug
140. lineage_taoshan-userdebug
55. lineage_hero2lte-userdebug
141. lineage_targa-userdebug

```

13. lineage_bacon-userdebug	56. lineage_herolte-userdebug
99. lineage_lux-userdebug	142. lineage_tenshi-userdebug
14. lineage_cancro-userdebug	57. lineage_hiaeuhl-userdebug
100. lineage_m216-userdebug	143. lineage_thea-userdebug
15. lineage_chagallwifi-userdebug	58. lineage_hiaeul-userdebug
101. lineage_m8d-userdebug	144. lineage_titan-userdebug
16. lineage_chaozu-userdebug	59. lineage_hltetmo-userdebug
102. lineage_m8-userdebug	145. lineage_tomato-userdebug
17. lineage_chel0-userdebug	60. lineage_hlte-userdebug
103. lineage_mako-userdebug	146. lineage_tsubasa-userdebug
18. lineage_cherry-userdebug	61. lineage_huashan-userdebug
104. lineage_maserati-userdebug	147. lineage_tulip-userdebug
19. lineage_clark-userdebug	62. lineage_hydrogen-userdebug
105. lineage_merlin-userdebug	148. lineage_umts_spyder-userdebug
20. lineage_condor-userdebug	63. lineage_i9100-userdebug
106. lineage_mido-userdebug	149. lineage_us996-userdebug
21. lineage_crackling-userdebug	64. lineage_i9300-userdebug
107. lineage_mint-userdebug	150. lineage_us997-userdebug
22. lineage_d800-userdebug	65. lineage_ido-userdebug
108. lineage_n1awifi-userdebug	151. lineage_v1awifi-userdebug
23. lineage_d801-userdebug	66. lineage_ivy-userdebug
109. lineage_n2awifi-userdebug	152. lineage_v400-userdebug
24. lineage_d802-userdebug	67. lineage_jag3gds-userdebug
110. lineage_n3-userdebug	153. lineage_v410-userdebug
25. lineage_d803-userdebug	68. lineage_jagnm-userdebug
111. lineage_n5100-userdebug	154. lineage_v480-userdebug
26. lineage_d850-userdebug	69. lineage_jalebi-userdebug
112. lineage_n5110-userdebug	155. lineage_v500-userdebug
27. lineage_d851-userdebug	70. lineage_jfltevwz-userdebug
113. lineage_n5120-userdebug	156. lineage_v521-userdebug
28. lineage_d852-userdebug	71. lineage_jfltexx-userdebug
114. lineage_nx512j-userdebug	157. lineage_vegetalte-userdebug
29. lineage_d855-userdebug	72. lineage_k3gxx-userdebug
115. lineage_odin-userdebug	158. lineage_victara-userdebug
30. lineage_deb-userdebug	73. lineage_karin-userdebug
116. lineage_onyx-userdebug	159. lineage_vs995-userdebug
31. lineage_dogo-userdebug	74. lineage_karin_windy-userdebug
117. lineage_osprey-userdebug	160. lineage_w5-userdebug
32. lineage_ether-userdebug	75. lineage_kccat6-userdebug
118. lineage_otus-userdebug	161. lineage_w7-userdebug
33. lineage_f400-userdebug	76. lineage_kenzo-userdebug
119. lineage_P024-userdebug	162. lineage_xt897-userdebug
34. lineage_falcon-userdebug	77. lineage_kingdom-userdebug
120. lineage_paella-userdebug	163. lineage_yuga-userdebug
35. lineage_find7s-userdebug	78. lineage_kipper-userdebug
121. lineage_peregrine-userdebug	164. lineage_Z00L-userdebug
36. lineage_find7-userdebug	79. lineage_kiwi-userdebug
122. lineage_piccolo-userdebug	165. lineage_Z00T-userdebug
37. lineage_flounder_lte-userdebug	80. lineage_klimtwifi-userdebug
123. lineage_pme-userdebug	166. lineage_zeroftexx-userdebug
38. lineage_flounder-userdebug	81. lineage_kltechnduo-userdebug

Last update:

2025/12/11 15:00 android:device-samsung-galaxy-s5-sm-g900f <https://looper.de/wiki/doku.php?id=android:device-samsung-galaxy-s5-sm-g900f>

124. lineage_pollux-userdebug	167. lineage_zeroltexx-userdebug
39. lineage_flo-userdebug	82. lineage_kltechn-userdebug
125. lineage_pollux_windy-userdebug	168. m_e_arm-userdebug
40. lineage_FP2-userdebug	83. lineage_klteduos-userdebug
126. lineage_r5-userdebug	169. mini_emulator_arm64-userdebug
41. lineage_g2m-userdebug	84. lineage_kltedv-userdebug
127. lineage_r7plus-userdebug	170. mini_emulator_x86_64-userdebug
42. lineage_gohan-userdebug	85. lineage_kltekd-di-userdebug
128. lineage_r7sf-userdebug	171. mini_emulator_x86-userdebug
43. lineage_gts210ltexx-userdebug	86. lineage_kltekor-userdebug
129. lineage_s2-userdebug	

Which would you like? [aosp_arm-eng] 88

Jetzt kann gebaut werden:

~# brunch

— [Steffen Bornemann](#) 08.11.2018

[Android](#), [Samsung](#), [Galaxy](#), [SM-G900F](#), [TWRP](#), [LineageOS](#), [Magisk](#), [Recovery](#), [PIT](#), [Stock-ROM](#), [Image](#), [CustomROM](#), [GIT](#)

From:

<https://looper.de/wiki/> - **Linux4Ever**

Permanent link:

<https://looper.de/wiki/doku.php?id=android:device-samsung-galaxy-s5-sm-g900f>

Last update: **2025/12/11 15:00**

