



Snagboot: vendor-agnostic, open-source and developer- friendly recovery and reflashing tool

Thomas Petazzoni, Bootlin

FOSDEM 2026

© Copyright 2004-2026, Bootlin.

Creative Commons BY-SA 3.0 license.

Corrections, suggestions, contributions and translations are welcome!





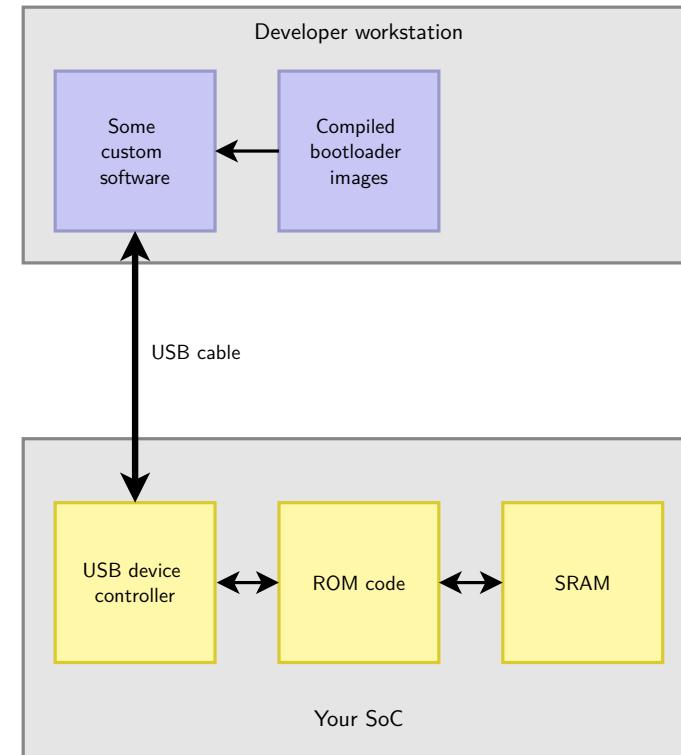
- ▶ Thomas Petazzoni
- ▶ CEO of *Bootlin*
 - Engineering company specialized in **Embedded Linux** and **Zephyr**
 - 28 people
 - Engineering services
 - Training services
 - Very strong open-source focus
 - We are **hiring**
- ▶ Former contributor to the *Linux kernel*: 900+ patches
- ▶ Co-maintainer and contributor to *Buildroot*





ROM code and recovery

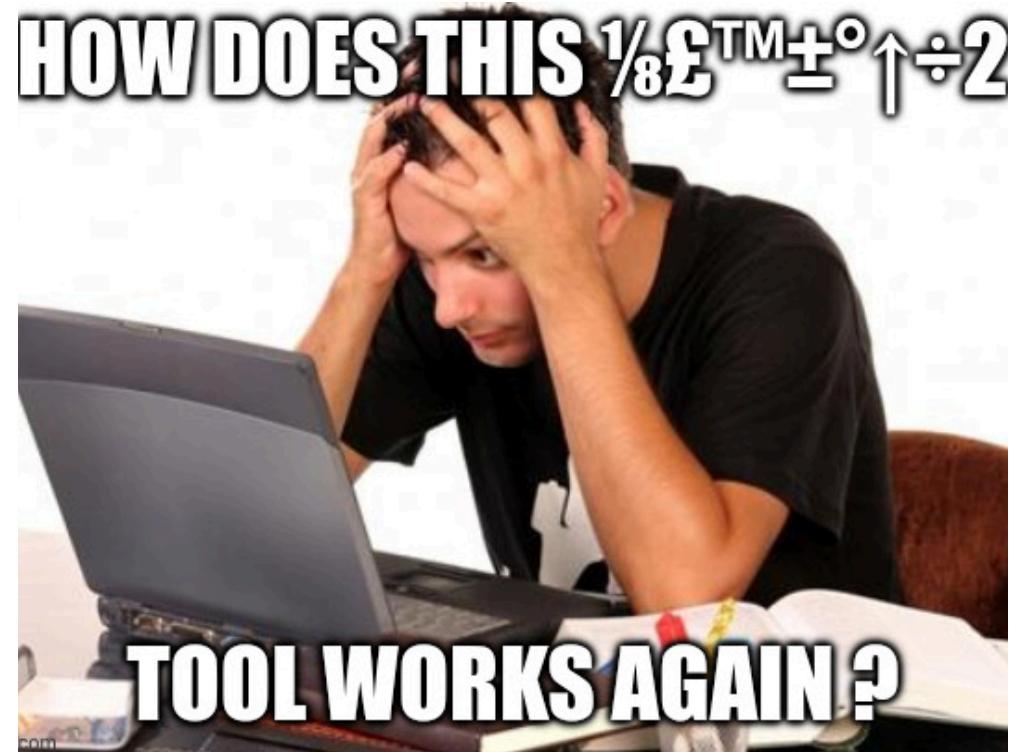
- ▶ Virtually all modern embedded SoCs integrate a ROM code
- ▶ Responsible for initial boot steps
- ▶ Generally provides a **recovery** mechanism
- ▶ Useful for
 - Factory flashing
 - Recovering bricked boards
- ▶ Typically uses a specific protocol on USB, sometimes UART or Ethernet
- ▶ Allows to send a binary payload into internal SRAM, and run it





Previous situation

- ▶ Each vendor has its **own** tool
 - NXP: `mfgtools` → uuu
 - ST: STM32 Cube Programmer
 - Microchip: SAM-BA
 - Rockchip: rkdeveloptool
 - etc.
- ▶ One tool per vendor is **annoying**
- ▶ Vendor tools are not always open-source
- ▶ Vendor tools are not always flexible/customizable





Enter Snagboot

- ▶ **One** project to rule them all
- ▶ **Vendor-agnostic**, supports **many** platforms
- ▶ **Open-source**: GPLv2
- ▶ **Hacker-friendly**: fully implemented in Python
- ▶ **Three** components
 - snagrecover
 - snagflash
 - snagfactory
- ▶ Works on Linux and Windows
- ▶ Developed and maintained by Bootlin engineer
Romain Gantois
- ▶ <https://github.com/bootlin/snagboot>
- ▶ <https://snagboot.readthedocs.io>
- ▶ Allwinner sunxi
- ▶ STMicroelectronics STM32MP
- ▶ Microchip SAMA5
- ▶ NXP i.MX6, i.MX7, i.MX8, i.MX9
- ▶ Texas Instruments AM335x, AM6x
- ▶ Xilinx/AMD Zynq UltraScale+
- ▶ Intel Keembay
- ▶ Broadcom BCM2711 (RPi 4) and BCM2712 (RPi 5)
- ▶ AMLogic: many SoCs supported



- ▶ Performs the recovery step: pushing a working bootloader on the target
 - Typically U-Boot
 - Bootloader is then expected to expose a USB gadget such as fastboot, UMS or DFU

```
$ cat fw.yaml
paths-relative-to: THIS_FILE

tiboot3:
  path: tiboot3_evm.bin
tispl:
  path: tispl_evm.bin
u-boot:
  path: u-boot_evm.img
```

```
$ snagrecover -s am625 -f fw.yaml
2026-01-28 17:26:42,257 [INFO] Starting recovery of am625
board
2026-01-28 17:26:42,260 [INFO] Installing firmware tiboot3
[...]
2026-01-28 17:26:44,082 [INFO] Installing firmware tispl
[...]
2026-01-28 17:26:45,648 [INFO] Installing firmware u-boot
[...]
2026-01-28 17:26:52,621 [INFO] Done recovering am625 board
$
```



- ▶ Performs the flashing step
- ▶ Assumes “something” is running on the target
 - Using **snagrecover** or some other mean
 - Typically U-Boot
 - Exposes Fastboot, USB Mass Storage or DFU

```
$ cat flash.cmd
set fb-addr 0xd0000000
set target mmc0

flash "random-1k.bin" 0 hwpart 1
flash "random-1M.bin" 1024
flash "random-100M.bin" 2048
flash "random-1G.bin.bz2" 2048

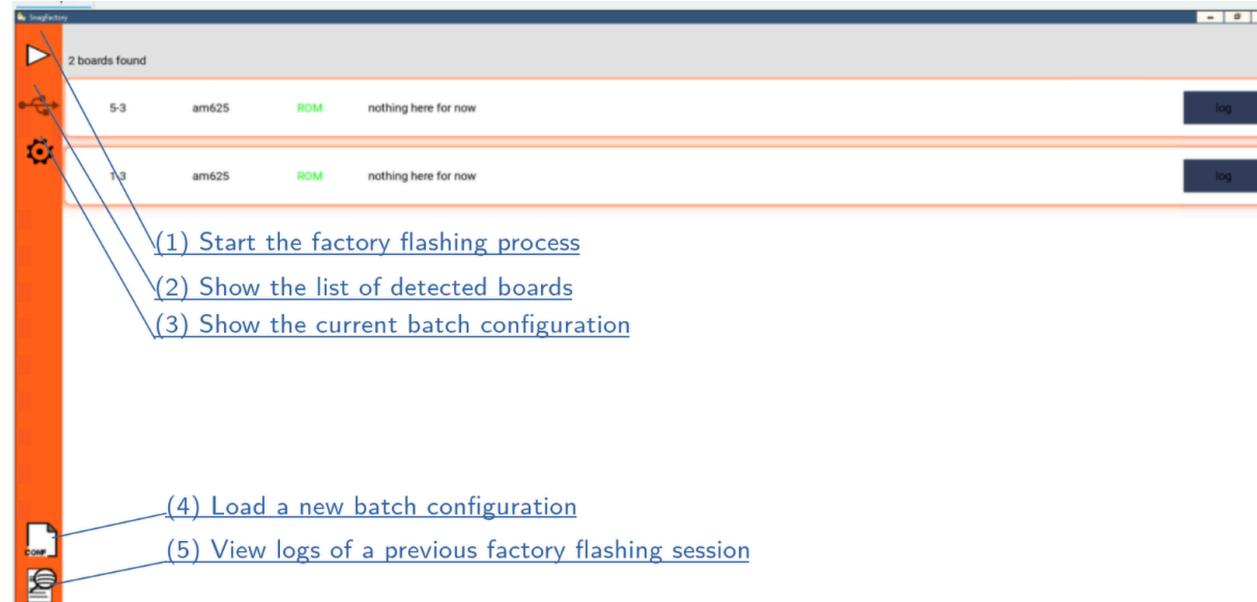
set target nor0
set eraseblk-size 0x40000

run oem_run:mtd list
flash "random-1k.bin" 0x40000
```

```
$ snagflash -P fastboot-uboot -p 0451:6165 -I flash.cmd
2026-01-28 17:27:04,916 [INFO] Running snagflash using
protocol fastboot-uboot
[...]
2026-01-28 17:27:04,929 [INFO] running command flash
"random-1k.bin" 0 hwpart 1
[...]
2026-01-28 17:27:05,321 [INFO] flashed 4096/? bytes
[...]
2026-01-28 17:29:14,805 [INFO] Flashing file random-1k.bin
2026-01-28 17:29:14,805 [INFO] Flashing to MTD device...
2026-01-28 17:29:15,639 [INFO] fastboot OKAY
2026-01-28 17:29:16,083 [INFO] flashed 4096/? bytes
```



- ▶ Factory flashing
- ▶ Builds on top of snagrecover/snagflash
- ▶ Multiple boards in parallel
- ▶ GUI oriented for operators





What to remember ?



UUU, STM32
Cube Programmer,
SAM-BA,
rkdevelopool,
sunxi-fel, etc.

Snagboot