7/LAB

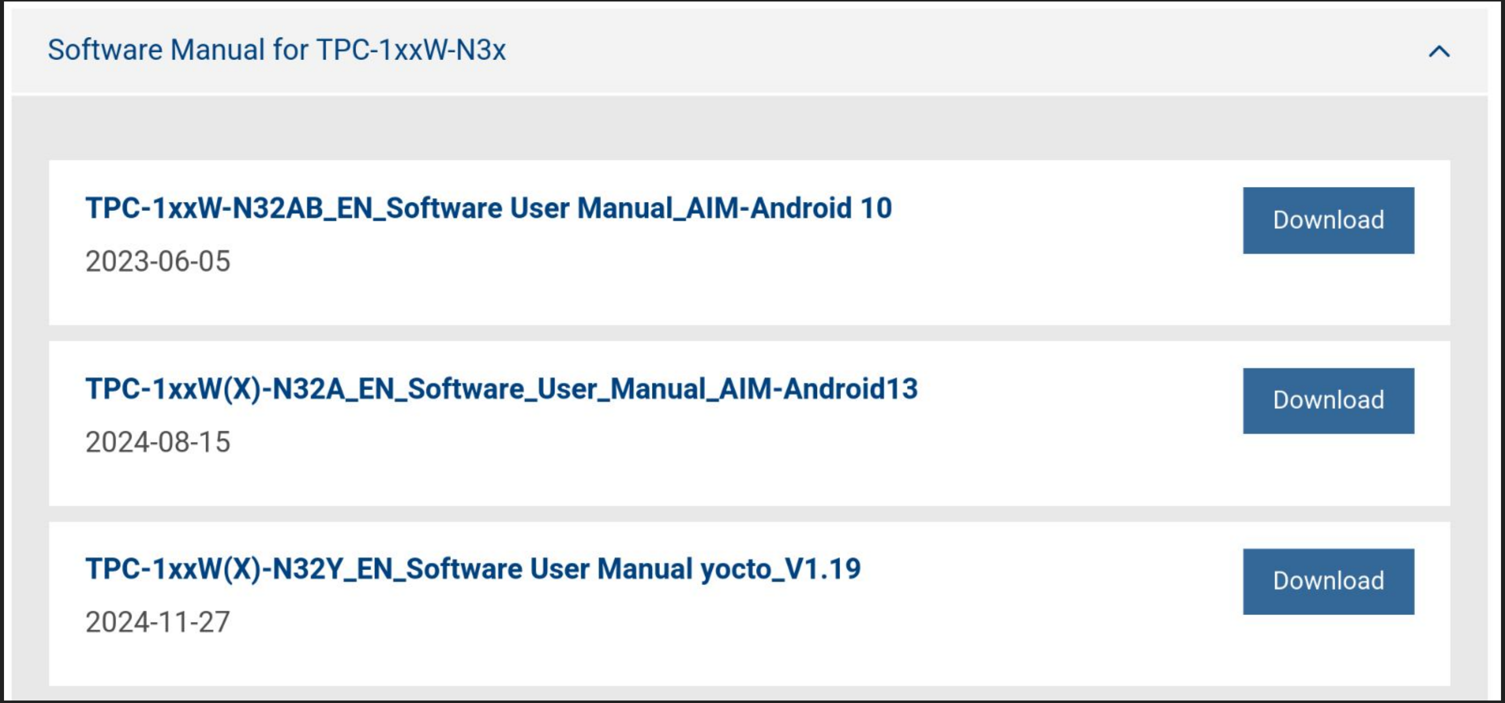
Keeping Android up to date when the OEM fails to

Michael Zimmermann
sevenlab engineering GmbH

https://sevenlab.de/2025/10/09/diy-software/

Advantech TPC-107W

Why did we do this?



Software Manual for TPC-1xxW-N3x

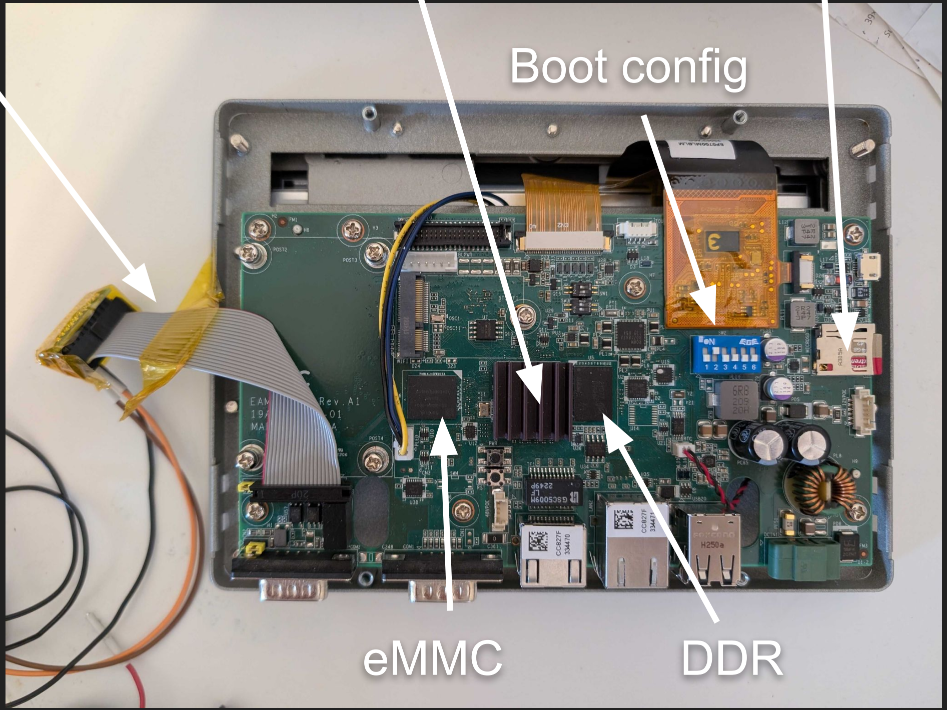
TPC-1xxW-N32AB_EN_Software User Manual_AIM-Android 10
2023-06-05
Download

TPC-1xxW(X)-N32A_EN_Software_User_Manual_AIM-Android13
2024-08-15
Download

TPC-1xxW(X)-N32Y_EN_Software User Manual yocto_V1.19
2024-11-27
Download

Advantech TPC-107W

2x serial ports (1x CAN)
2x ethernet (1GBit, PoE)
2x USB 2.0



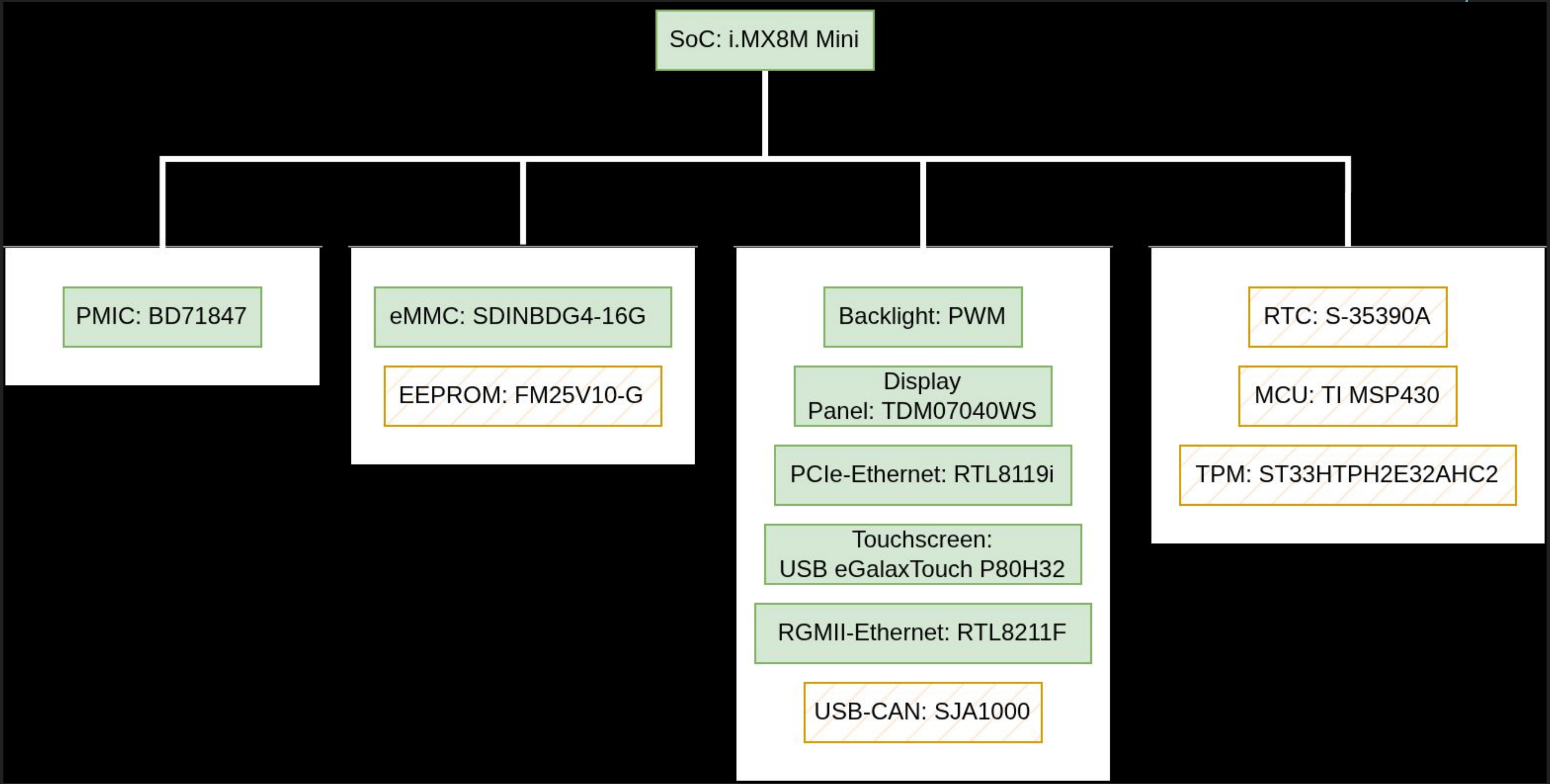
7/LAB

What did Advantech change in the code?

- Add U-Boot and Linux **devicetrees**
- Add new **drivers**: Ethernet, external watchdog, USB-CAN, eeprom, panel
- New **U-Boot** board with custom hardware init sequence.
- Minor hacks and workarounds in Linux drivers.
- A lot of **Android** changes unrelated to the hardware.

Categories

- **Support for hardware which we need**
- Support for hardware we don’t need
- Generic customization


SoC: i.MX8M Mini
PMIC: BD71847
eMMC: SDINBDG4-16G
EEPROM: FM25V10-G
Backlight: PWM
Display Panel: TDM07040WS
PCIe-Ethernet: RTL8119i
Touchscreen: USB eGalaxTouch P80H32
RGMII-Ethernet: RTL8211F
USB-CAN: SJA1000
RTC: S-35390A
MCU: TI MSP430
TPM: ST33HTPH2E32AHC2

Where to start?

Goal: Boot to lock screen as soon as possible.

TODOs

- Bootloader
- Kernel
- Android

Our strategy

- Keep Advantechs Bootloader
- Compile Android+Kernel for NXP devkit, Kernel devicetree from Advantech.

The first boot to the lock screen

- Devicetree and cmdline hacks to boot with the old Bootloader
- Replace `panel-simple.c` with Advantechs version.
- Enable more drivers in `imx8mm_gki.fragment`
- Disable AVB (Android Verified Boot).

Issues we had

- Android partition mounting and verification was hard to debug. Had to do print-debugging.
- Advantech had a wrong debug UART pin config.

Port the latest U-Boot

- Different pin config
- Different PMIC, Panel, NOR
- Remove unsupported features: USB-C, USB-OTG
- Add GPIO-based init sequence for several devices on the board.
- Copy ethernet mac address from SPI flash to device tree.

Cleanup 🧹

- Rename Files and strings: evk_8mm -> tpc_1xx
- Put Android device into separate git repository
- Remove unused features: devkit variants, camera, audio (partially)
- Make imx-mkimage and imx-make.sh support the new device

Improving the Code

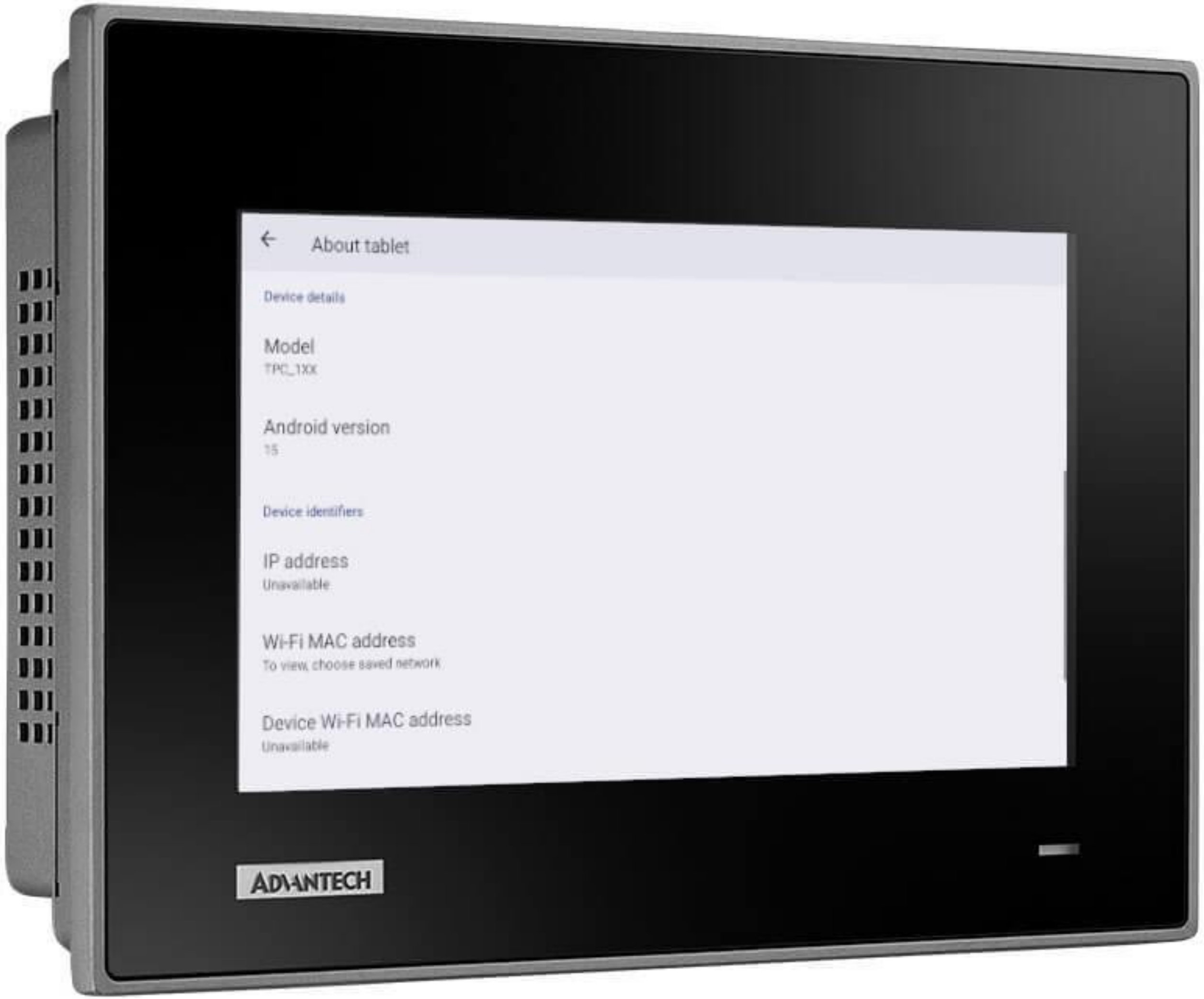
- Panel: Write separate driver, port it to uboot, patch “compatible” only.
- Rewrite Linux/U-Boot devicetree based on NXP devkit.
- AVB and HABv4.

NXP HABv4

- Can be used with publicly available information.
- Requires a eMMC with RPMB support (we were lucky)
- **Set `CONFIG_AUTO_SET_RPMB_KEY=y` in U-Boot.**

Issues we had

- We provisioned RPMB with HABv4 disabled.
- There was no easy way to automate signing during development.



About tablet

Device details

Model
TPC_1XX

Android version
15

Device identifiers

IP address
Unavailable

Wi-Fi MAC address
To view, choose saved network

Device Wi-Fi MAC address
Unavailable

New files in Bootloader

- arch/arm/dts/imx8mm-eamb9918*.dts*
- board/advantech/imx8mm_eamb9918/
- configs/imx8mm_eamb9918_android*_defconfig
- drivers/video/panel-advantech.c
- include/configs/imx8mm_eamb9918*.h

Modified files in Bootloader

- arch/arm/dts/Makefile
- arch/arm/mach-imx/imx8m/Kconfig
- arch/arm/mach-imx/imx8m/soc.c
- arch/riscv/lib/fdt_fixup.c
- common/board_f.c
- drivers/fastboot/fb_fsl/fb_fsl_boot.c
- drivers/video/Kconfig
- drivers/video/Makefile
- dts/Kconfig
- include/init.h
- …

7/LAB

New files in Kernel

- arch/arm64/boot/dts/freescale/
  - imx8mm-eamb9918-android.dtsi
  - imx8mm-eamb9918.dts
  - imx8mm-eamb9918.dtsi
- drivers/gpu/drm/panel/panel-advantech.c

Modified files in Kernel

- arch/arm64/boot/dts/freescale/Makefile
- arch/arm64/configs/imx8mm_gki.fragment
- drivers/gpu/drm/panel
  - Kconfig
  - Makefile

Should you do this?

Only if:

- The SoC vendor provides a complete, **publicly available maintained BSP** for their devkit.
- The Board schematics can be understood well enough from the existing code or through simple hardware analysis.
- **Closed source parts don’t need to be modified** for the hardware features you need.