./kernel concepts

Android 12 BSP for MSC SM2S-IMX8M Modules

Getting Started - November 20, 2023

Description

Android BSP for SM2S-IMX8M SOMs version 2.1.5- based on imx-android 12.0.0_1.0.0 (Android 12).

Package Contents

- README
  contains quick instructions
- bsp_init.sh
  script to download and install the BSP tree on the build host
- bsp_build.sh
  script to build the BSP
- patches.tar.bz2
  Patchset against imx-android
- required_packages_ubuntu_focal
  list of required packages for Ubuntu Focal Fossa

Supported Hardware

- SOMs
  - SM2S-IMX8PLUS-QC-14N0740I
  - SM2S-IMX8MINI-QC-14N0261I
- Baseboards
  - SM2-MB-EP1-001
- Displays
  - MSC DIS-AM1280800-101-KIT (lvds)

Supported Features

- WiFi (802.11ac/b/g/n)
- Bluetooth/BLE

> HCI device suspending had to be disabled in order to get the driver working. Expect additional power consumption!

- Ethernet
- USB Host
- USB Client

> – For user-builds ADB has to be enabled in userspace before it becomes available. [5]
> – Data transfer functions like e.g. mtp have to be explicitly selected in userspace under *Settings->Connected devices->USB*

- MSC SMARC MIPI Camera Kit001
- Audio Output and Input via SGTL5000 sound codec
- SDHC-Slot

- Volume Control via LID and SLEEP Buttons

> Disabled for IMX8PLUS Modules due to missing PullUp

Prerequisites

imx-android

This BSP requires the imx-android tarball from [1]

Toolchain

Certain parts of the original imx-android require the GNU ARM toolchain [2] to compile.

WiFi drivers

We also require access to git repository at

```
git+ssh://gitolite@msc-git02.msc-ge.com:9418/thirdparty/linux-sdio-driver-209a .
```

Please confer with Avnet Embedded Support about this.

OS

This BSP was tested on Ubuntu 20.04 Focal Fossa with the following additional packages:

```
uuid uuid-dev zlib1g-dev liblz-dev liblzo2-2 liblzo2-dev lzop git-core curl u-boot-tools \
mtd-utils openjdk-8-jdk device-tree-compiler gdisk m4 libz-dev bison \
flex libssl-dev rsync android-sdk-libsparse-utils android-sdk-ext4-utils \
libncurses5:i386 libncurses5 libncurses5-dev python-is-python3 python bc cpio zip
```

> libncurses5:i386 is only available in universe repositories, which may not be active:
>
> ```
> sudo add-apt-repository universe
> ```

Additionally, the Universal Update Utility [3] is needed.

Build Hardware

AOSP and therefore imx-android and this BSP contain a lot of code that needs to be compiled. A rather potent machine is advisable. For comparison: A full build of this BSP from scratch takes about 140 Minutes on a dual Xeon E5-2650 v2 (16C/32T). 1GiB of RAM per logical CPU is recommended. This BSP requires up to 310GiB of drive space for source tree, intermediates and results.

Initialization

> In case you have not configured git yet (i.e. fresh installation) you need to supply some information:
>
> ```
> git config --global user.email your mail address
> git config --global user.name your name
> ```

To install and set up, unzip the BSP tarball to a folder of your choice and place the imx-android package and the GNU ARM toolchain in this folder. Then use the bsp_init.sh script to download and prepare the sources:

```
./bsp_init.sh -r
```

This will install imx-android and the GNU ARM toolchain to PWD and then patch imx-android to support IMX8M SMARC modules. Depending on your internet connection this may take a while as imx-android installation involves cloning a lot of git repositories over the internet.

The script accepts several options:

- -r | –localrepo
  Do not expect repo tool to be in PATH, download it to PWD and use it from there.

Building

Scripted Build

To build the binaries for a device run

```
./bsp_build.sh
```

This script takes several options:

- -t | –type
  SOM-Type. Default: sm2s
- -m | –model
  SOM-Model. Default: imx8plus
- -d | –display
  Display to run on. Default: ama121a01 (MSC DIS-AM1280800-101-KIT)
- -v | –variant
  SOM-variant. Default: 14N0740I
- -j | –jobs
  Number of concurrent build threads to run. Default: 1
- -b | –buildvariant
  Build variant of Android (eng/user/userdebug). Default: user
- -c | –tc_path
  Path to GNU ARM Toolchain. Default: $PWD/gcc-arm-8.3-2019.03-x86_64-aarch64-linux-gnu
  The default is correct for installation via bsp_init.sh.

Manual Build by example of IMX8PLUS

1. change into build directory

   ```
   $ cd android_build/
   ```

2. source build environment

   ```
   $ . build/envsetup.sh
   ```

3. select device using launch command

   ```
   $ lunch

   You're building on Linux

   Lunch menu... pick a combo:
   1. aosp_arm-eng
   2. aosp_arm64-eng
   3. aosp_barbet-userdebug
   4. aosp_blueline-userdebug
   ```

```
5. aosp_blueline_car-userdebug
6. aosp_bonito-userdebug
7. aosp_bonito_car-userdebug
8. aosp_bramble-userdebug
9. aosp_bramble_car-userdebug
10. aosp_car_arm-userdebug
11. aosp_car_arm64-userdebug
12. aosp_car_x86-userdebug
13. aosp_car_x86_64-userdebug
14. aosp_cf_arm64_auto-userdebug
15. aosp_cf_arm64_phone-userdebug
16. aosp_cf_x86_64_foldable-userdebug
17. aosp_cf_x86_64_pc-userdebug
18. aosp_cf_x86_64_phone-userdebug
19. aosp_cf_x86_64_tv-userdebug
20. aosp_cf_x86_auto-userdebug
21. aosp_cf_x86_phone-userdebug
22. aosp_cf_x86_tv-userdebug
23. aosp_coral-userdebug
24. aosp_coral_car-userdebug
25. aosp_crosshatch-userdebug
26. aosp_crosshatch_car-userdebug
27. aosp_crosshatch_vf-userdebug
28. aosp_flame-userdebug
29. aosp_flame_car-userdebug
30. aosp_redfin-userdebug
31. aosp_redfin_car-userdebug
32. aosp_redfin_vf-userdebug
33. aosp_sargo-userdebug
34. aosp_sargo_car-userdebug
35. aosp_sunfish-userdebug
36. aosp_sunfish_car-userdebug
37. aosp_trout_arm64-userdebug
38. aosp_trout_x86-userdebug
39. aosp_x86-eng
40. aosp_x86_64-eng
41. arm_krait-eng
42. arm_v7_v8-eng
43. armv8-eng
44. armv8_cortex_a55-eng
45. armv8_kryo385-eng
46. beagle_x15-userdebug
47. beagle_x15_auto-userdebug
48. car_x86_64-userdebug
49. db845c-userdebug
50. evk_6sl-user
51. evk_6sl-userdebug
52. evk_7ulp-user
53. evk_7ulp-userdebug
54. evk_7ulp_revb-user
55. evk_7ulp_revb-userdebug
56. evk_8mm-user
57. evk_8mm-userdebug
58. evk_8mm_drm-user
59. evk_8mm_drm-userdebug
60. evk_8mn-user
61. evk_8mn-userdebug
62. evk_8mp-user
63. evk_8mp-userdebug
64. evk_8mq-user
65. evk_8mq-userdebug
66. evk_8mq_drm-user
67. evk_8mq_drm-userdebug
68. evk_8ulp-user
69. evk_8ulp-userdebug
```

```
70. fuchsia_arm64-eng
71. fuchsia_x86_64-eng
72. gsi_car_arm64-userdebug
73. gsi_car_x86_64-userdebug
74. hikey-userdebug
75. hikey64_only-userdebug
76. hikey960-userdebug
77. hikey960_tv-userdebug
78. hikey_tv-userdebug
79. mek_8q-user
80. mek_8q-userdebug
81. mek_8q_car-user
82. mek_8q_car-userdebug
83. mek_8q_car2-user
84. mek_8q_car2-userdebug
85. pixel3_mainline-userdebug
86. poplar-eng
87. poplar-user
88. poplar-userdebug
89. qemu_trusty_arm64-userdebug
90. sabreauto_6q-user
91. sabreauto_6q-userdebug
92. sabresd_6dq-user
93. sabresd_6dq-userdebug
94. sabresd_6dq_car-user
95. sabresd_6dq_car-userdebug
96. sabresd_6sx-user
97. sabresd_6sx-userdebug
98. sabresd_7d-user
99. sabresd_7d-userdebug
100. sdk_car_arm-userdebug
101. sdk_car_arm64-userdebug
102. sdk_car_x86-userdebug
103. sdk_car_x86_64-userdebug
104. silvermont-eng
105. sm2s_imx8mini_13N4200I_lvds0_native_single_mode_ama121a01-user
106. sm2s_imx8mini_13N4200I_lvds0_native_single_mode_ama121a01-userdebug
107. sm2s_imx8mini_14N0261I_lvds0_native_single_mode_ama121a01-user
108. sm2s_imx8mini_14N0261I_lvds0_native_single_mode_ama121a01-userdebug
109. sm2s_imx8plus_14N0740I_lvds0_native_single_mode_ama121a01-user
110. sm2s_imx8plus_14N0740I_lvds0_native_single_mode_ama121a01-userdebug
111. uml-userdebug
112. yukawa-userdebug
113. yukawa_sei510-userdebug

Which would you like? [aosp_arm-eng] 109
10:09:27 Build sandboxing disabled due to nsjail error.
10:09:28 Build sandboxing disabled due to nsjail error.

============================================
PLATFORM_VERSION_CODENAME=REL
PLATFORM_VERSION=12
TARGET_PRODUCT=sm2s_imx8plus_14N0740I_lvds0_native_single_mode_ama121a01
TARGET_BUILD_VARIANT=user
TARGET_BUILD_TYPE=release
TARGET_ARCH=arm64
TARGET_ARCH_VARIANT=armv8-a
TARGET_CPU_VARIANT=cortex-a53
TARGET_2ND_ARCH=arm
TARGET_2ND_ARCH_VARIANT=armv7-a-neon
TARGET_2ND_CPU_VARIANT=cortex-a9
HOST_ARCH=x86_64
HOST_2ND_ARCH=x86
HOST_OS=linux
HOST_OS_EXTRA=Linux-6.5.0-gentoo-x86_64-Ubuntu-20.04.6-LTS
```

```
HOST_CROSS_OS=windows
HOST_CROSS_ARCH=x86
HOST_CROSS_2ND_ARCH=x86_64
HOST_BUILD_TYPE=release
BUILD_ID=SQ1A.220105.002
OUT_DIR=out
PRODUCT_SOONG_NAMESPACES=device/generic/goldfish device/generic/goldfish-opengl external/mesa3d
    vendor/nxp-opensource/imx/power hardware/google/pixel hardware/google/camera vendor/nxp-
    opensource/imx/camera vendor/partner_gms
============================================
```

(replace 109 with correct index number for your board)

4. build i.MX specific parts (“vendor” parts)

```
AARCH64_GCC_CROSS_COMPILE="$(realpath ../gcc-arm-8.3-2019.03-x86_64-aarch64-linux-gnu)"/\
bin/aarch64-linux-gnu- \
./imx-make.sh kernel bootloader bootimage vendorimage dtboimage -j1
```

5. build AOSP

```
make -j1 droid fastboot
```

Cleaning

To clean artifacts for a specific device:

```
rm -rf android_build/out/target/product/${your device}/*
```

(Replace ${your device} with your devices name. E.g. “sm2s_imx8mini” or “sm2s_imx8plus”)

To also clean artifacts that are not device specific:

```
rm -rf android_build/out/target/*
```

To start the next build completely from scratch:

```
rm -rf android_build/out/*
```

Installation on target

Connect the target via USB to the build host, set the FORCE_RECOVERY switch on the target and apply power to it.

> Make sure to reset all other switches.

Change into the output directory, e.g. from the directory where bsp_build.sh resides:

```
cd android_build/out/target/product/sm2s_imx8plus/
```

And run uuu on uuu.lst:

```
sudo uuu uuu.lst
```

Reset the FORCE_RECOVERY switch and power cycle the target to boot.

Usage

Customization

Adding your own device by example of an IMX8PLUS based device

To add a device named “my_own_device” do the following:

1. navigate to android_build/device/nxp/imx8m/sm2s_imx8plus/

   ```
   cd android_build/device/nxp/imx8m/sm2s_imx8plus/
   ```

2. Fire up an editor of your choice to create the file “my_own_device.mk”

   ```
   nano my_own_device.mk
   ```

3. Select your display connection by adding the following lines:

   ```
   MSC_DISPLAY_STR := lvds0_native_single_mode
   ```

4. Select your display by adding the following lines:

   ```
   MSC_DISPLAY := ama121a01
   TARGET_BOARD_DTO_CONFIG += overlay-lvds0-ama-121a01.dtb
   ```

5. Select your SMARC-module by adding the following lines:

   ```
   include device/fsl/imx8m/sm2s_imx8plus/sm2s_imx8plus_14N0740I.mk
   include device/fsl/imx8m/sm2s_imx8plus/sm2s_imx8plus_common.mk
   ```

6. include SMARC general definitions by adding the following line:

   ```
   include device/fsl/imx8m/sm2s_common/sm2s.mk
   ```

7. set your product’s name, model and device by adding the following lines:

   ```
   PRODUCT_NAME := ${MSC_SOM_FAM}_${MSC_SOM_TYPE}_${MSC_SOM_VARIANT}_${MSC_DISPLAY_STR}_${
       MSC_DISPLAY}
   PRODUCT_MODEL := ${MSC_SOM_FAM}_${MSC_SOM_TYPE}_${MSC_SOM_VARIANT}_${MSC_DISPLAY_STR}_${
       MSC_DISPLAY}
   PRODUCT_DEVICE := ${MSC_SOM_FAM}_${MSC_SOM_TYPE}
   ```

8. save the file and exit the editor.
9. navigate into parent directory

   ```
   cd ..
   ```

10. edit AndroidProducts.mk:

    ```
    nano AndroidProducts.mk
    ```

11. under “PRODUCT_MAKEFILES :=
    ” add the following line:

    ```
    $(LOCAL_DIR)/sm2s_imx8plus/my_own_device.mk \
    ```

12. under “COMMON_LUNCH_CHOICES :=
    ” add the following lines:

    ```
    my_own_device-user \
    my_own_device-userdebug
    ```

    It should now look like this:

```
PRODUCT_MAKEFILES := \
$(LOCAL_DIR)/evk_8mq/evk_8mq.mk \
$(LOCAL_DIR)/evk_8mq/evk_8mq_drm.mk \
$(LOCAL_DIR)/evk_8mm/evk_8mm.mk \
$(LOCAL_DIR)/evk_8mm/evk_8mm_drm.mk \
$(LOCAL_DIR)/evk_8mn/evk_8mn.mk \
$(LOCAL_DIR)/evk_8mp/evk_8mp.mk \
$(LOCAL_DIR)/sm2s_imx8plus/sm2s_imx8plus_14N0740I_lvds0_native_single_mode_ama121a01.mk \
$(LOCAL_DIR)/sm2s_imx8mini/sm2s_imx8mini_14N0261I_lvds0_native_single_mode_ama121a01.mk \
$(LOCAL_DIR)/sm2s_imx8mini/sm2s_imx8mini_13N4200I_lvds0_native_single_mode_ama121a01.mk \
$(LOCAL_DIR)/sm2s/my_own_device.mk

COMMON_LUNCH_CHOICES := \
evk_8mq-user \
evk_8mq-userdebug \
evk_8mq_drm-user \
evk_8mq_drm-userdebug \
evk_8mm-user \
evk_8mm-userdebug \
evk_8mm_drm-user \
evk_8mm_drm-userdebug \
evk_8mn-user \
evk_8mn-userdebug \
evk_8mp-user \
evk_8mp-userdebug \
sm2s_imx8plus_14N0740I_lvds0_native_single_mode_ama121a01-user \
sm2s_imx8plus_14N0740I_lvds0_native_single_mode_ama121a01-userdebug \
sm2s_imx8mini_13N4200I_lvds0_native_single_mode_ama121a01-user \
sm2s_imx8mini_13N4200I_lvds0_native_single_mode_ama121a01-userdebug \
sm2s_imx8mini_14N0261I_lvds0_native_single_mode_ama121a01-user \
sm2s_imx8mini_14N0261I_lvds0_native_single_mode_ama121a01-userdebug \
m2s_imx8mini_14N0261I_lvds0_native_single_mode_ama121a01-userdebug \
my_own_device-user \
my_own_device-userdebug
```

Mind the backslashes ("\")!

13. Build the system as described in Manual Build by example of IMX8PLUS

Known Issues

Wireless

Bluetooth and Wireless LAN connectivity seems to be impaired by the heatsink ontop of the SOM. We recommend using a suitable Antenna.

Display

When switching between displays their respective pixel density has to be taken into account. The pixel density can be set via the *ro.sf.lcd_density* property [4].

USB Client

It has been observed that USB Client connection does not always work. Exact circumstances are currently unknown.

Further Reading

- WiFi and Bluetooth driver and firmware license agreement

  `android_build/vendor/msc/linux-sdio-driver-209a/HDW Software Use License Agreement Template.pdf`

- Freescale imx-android Documentation
  https://www.nxp.com/docs/en/supporting-information/android_Q{12.0.0_1.0.0}_docs.zip
- Android App Developer Documentation
  https://developer.android.com/docs

References

[1] imx-android 12.0.0_1.0.0 (NXP Login required): https://www.nxp.com/webapp/sps/download/license.jsp?colCode=12.0.0_1.0.0_ANDROID_SOURCE&appType=file1&DOWNLOAD_ID=null&

[2] gcc-arm toolchain: https://developer.arm.com/-/media/Files/downloads/gnu-a/8.3-2019.03/binrel/gcc-arm-8.3-2019.03-x86_64-aarch64-linux-gnu.tar.xz?revision=2e88a73f-d233-4f96-b1f4-d8b36e9bb0b9&hash=C6CE74D7E1189F53EAC2D9DB08AADE29

[3] Universal Update Utility (uuu): https://github.com/NXPmicro/mfgtools/releases

[4] AOSP documentation about dpi: https://source.android.com/docs/automotive/hmi#density

[5] HowTo enable Android Debug Bridge: https://developer.android.com/tools/adb#Enabling