**CM-AM62x / CX-AM62x Android13 Manual**

**Revision 1.3**

**2024. 01. 16**

[**CRZ Technology**](http://www.crz-tech.com/)

**Document History**

|  |  |  |
| --- | --- | --- |
| **Revision** | **Date** | **Change note** |
| **1.0** | **2023.08.04** | **Created** |
| **1.1** | **2023.09.05** | **Implemented sd card fusing** |
| **1.2** | **2023.12.18** | **Supported Android on SD** |
| **1.3** | **2024.01.16** | **Added figure of debug cable connection** |

**Index**

[1. 개요 4](#_Toc156313813)

[2. 호스트 빌드 환경 4](#_Toc156313814)

[3. Bootloader 빌드 4](#_Toc156313815)

[4. Kernel 빌드 4](#_Toc156313816)

[5. Android 빌드 4](#_Toc156313817)

[6. SD Boot 5](#_Toc156313818)

[7. eMMC Fusing 5](#_Toc156313819)

[8. LCD 지원 6](#_Toc156313820)

[9. Android on SD 6](#_Toc156313821)

# **개요**

이 문서에서는 CM-AM62x / CX-AM62x 보드를 위한 안드로이드 빌드 방법을 기술한다.

# **호스트 빌드 환경**

호스트 빌드 환경은 Ubuntu 22.04 LTS를 이용한다.

아래 링크에서 3가지 툴체인을 가져와서 /opt/에서 압축을 해제한다.

|  |
| --- |
| [arm tool chain](https://drive.google.com/drive/folders/1dgu3cJRKvY3eag_Gi7C5m_N6Z0v5uZvw?usp=sharing) |

아래 링크에서 AM62x 안드로이드 빌드용 소스를 가져와서 압축을 해제한다.

|  |
| --- |
| [am62x\_android\_20230905.tar.gz](https://drive.google.com/file/d/1edgVd-utclh0-SukJ3JC08a3vrmIufkw/view?usp=sharing) |

# **Bootloader 빌드**

bootloader를 빌드하기 위해서 아래 명령을 입력한다.

|  |
| --- |
| $ ./build.sh bootloader |

# **Kernel 빌드**

Linux kernel을 빌드하기 위해서 아래 명령을 입력한다.

|  |
| --- |
| $ ./build.sh kernel |

# **Android 빌드**

Android를 빌드하기 위해서 아래 명령을 입력한다.

|  |
| --- |
| $ ./build.sh android |

Bootloader / kernel / Android를 clean하기 위해서 아래 명령을 입력한다.

|  |
| --- |
| $ ./build.sh clean |

# **SD Boot**

uSD 카드(16GB 이상)를 uSD 카드 어댑터를 통하여 호스트 Ubuntu에 연결한다.

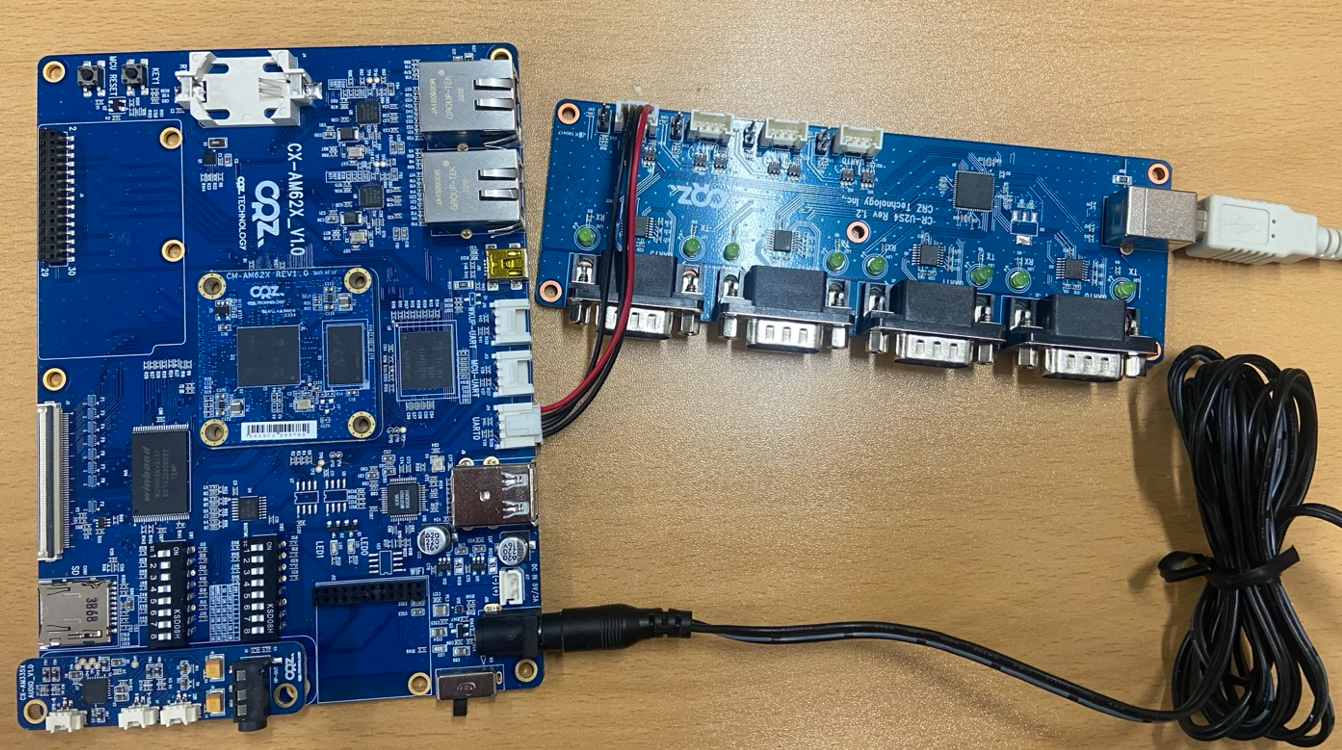
부팅가능한 uSD 카드를 만들기 위해서 image 폴더에서 “sudo ./create-sdcard\_ubuntu.sh”를 실행한다.

|  |
| --- |
| $ cd image  $ sudo ./create-sdcard\_ubuntu.sh |

SW1을 ON-ON-ON-OFF(1-2-3-4)로 설정한다.

uSD를 슬롯에 넣고 보드에 5V 파워를 인가하고 시리얼 터미널에 로그가 나오는지 확인한다.

Debug 케이블은 J14 UART0에 연결한다. 아래는 Debug 케이블 연결 사진이다.



PC와 연결시 COM 포트가 4개 생성이 되는데 로그가 나오는 포트가 있는지 확인한다.

# **eMMC Fusing**

SD boot를 확인한 uSD 카드를 다시 호스트에 연결한다.

아래 명령을 실행하여 eMMC boot를 가능하게 해 주는 uSD를 만든다.

|  |
| --- |
| $ cd ./image  $ sudo ./make\_auto\_android\_eMMC\_fuse\_SD.sh |

SW1을 ON-ON-ON-OFF(1-2-3-4)로 설정한다.

uSD를 슬롯에 넣고 보드에 5V 파워를 인가하고 로그인하면 eMMC에 android가 설치된다.

보드 파워를 끄고 uSD 카드를 제거한다.

SW1을 OFF-ON-ON-OFF(1-2-3-4)로 설정한다.

보드를 껐다가 켜서 Android boot가 되는지 확인한다.

# **LCD 지원**

innolux-at070tn92 / innolux-m236hjj-l31 / bsd101wx1-300 세가지 LCD를 지원할 수 있다.

Uboot에서 아래와 같은 명령으로 LCD를 선택한다.

dtbo\_index=0 /\* k3-am625-sk-innolux-at070tn92.dtbo \*/

|  |
| --- |
| => setenv dtbo\_index 0  => run bootcmd\_system |

dtbo\_index=1 /\* k3-am625-sk-innolux-m236hjj-l31.dtbo \*/

|  |
| --- |
| => setenv dtbo\_index 1  => run bootcmd\_system |

dtbo\_index=2 /\* k3-am625-sk-bsd101wx1-300.dtbo \*/

|  |
| --- |
| => setenv dtbo\_index 2  => run bootcmd\_system |

소스에서는 ti-bootloader-aosp\ti-u-boot\board\ti\am62x\am62x.env 에서 원하는 값으로 수정한다.

|  |
| --- |
| dtbo\_index=2  #if 0  dtbo\_index=0 /\* k3-am625-sk-innolux-at070tn92.dtbo \*/  dtbo\_index=1 /\* k3-am625-sk-innolux-m236hjj-l31.dtbo \*/  dtbo\_index=2 /\* k3-am625-sk-bsd101wx1-300.dtbo \*/  #endif |

# **Android on SD**

SD card에서 동작하는 Android를 빌드하기 위해서 아래 명령을 입력한다.

|  |
| --- |
| $ ./build.sh android sd |

uSD 카드(16GB 이상)를 uSD 카드 어댑터를 통하여 호스트 Ubuntu에 연결한다.

Android 부팅가능한 uSD 카드를 만들기 위해서 image 폴더에서 “sudo ./create-android-sdcard\_ubuntu.sh”를 실행한다.

|  |
| --- |
| $ cd image  $ sudo ./create-android-sdcard\_ubuntu.sh |

SW1을 ON-ON-ON-OFF(1-2-3-4)로 설정한다.

uSD를 슬롯에 넣고 보드에 5V 파워를 인가하고 Android boot가 되는지 확인한다.

eMMC에 Android가 이미 설치되어 있는 경우 Android boot가 실패할 수 있다.

uboot에서 eMMC에 설치된 Android를 지워주면 정상 부팅이 된다.

|  |
| --- |
| => mmc dev 0 0  => mmc erase 0 0x10000  => mmc dev 0 1  => mmc erase 0 0x10000 |