

EM-IMX8M-MINI Linux User Manual

V1.1



Boardcon Embedded Design

www.boardcon.com



Revision History

Ver	Description	Author	Date
V1.0	Initial version	Yang Jing	2019-11-19
V1.1	Modify testing	Zhou Lijun	2019-11-20

1. Introduction

1.1. About this Manual

This manual is intended to provide the user with an overview of the board and benefits, complete features specifications, and set up procedures. It contains important safety information as well.

1.2. Feedback and Update to this Manual

To help our customers make the most of our products, we are continually making additional and updated resources available on the Boardcon website (www.boardcon.com , www.armdesigner.com).

These include manuals, application notes, programming examples, and updated software and hardware. Check in periodically to see what's new!

When we are prioritizing work on these updated resources, feedback from customers is the number one influence, If you have questions, comments, or concerns about your product or project, please no hesitate to contact us at support@armdesigner.com.

1.3. Limited Warranty

Boardcon warrants this product to be free of defects in material and workmanship for a period of one year from date of buy. During this warranty period Boardcon will repair or replace the defective unit in accordance with the following process:

A copy of the original invoice must be included when returning the defective unit to Boardcon. This limited warranty does not cover damages resulting from lighting or other power surges, misuse, abuse, abnormal conditions of operation, or attempts to alter or modify the function of the product.

This warranty is limited to the repair or replacement of the defective unit. In no event shall Boardcon be liable or responsible for any loss or damages, including but not limited to any lost profits, incidental or consequential damages, loss of business, or anticipatory profits arising from the use or inability to use this product.

Repairs make after the expiration of the warranty period are subject to a repair charge and the cost of return shipping. Please contact Boardcon to arrange for any repair service and to obtain repair charge information.

Content

1 EM-IMX8M-MINI Introduction	4
1.1 Summary	4
1.2 Processor Features	4
1.3 EM-IMX8M-MINI specifications	5
2. Compiler Environment	5
2.1 Vmware10.0+ubuntu16.04	5
2.2 Install Tools	6
3. Compile the Source	8
3.1 Compile Uboot	8
3.2 Compile Kernel	8
3.3 Compile rootfs	10
4 Install Serial Terminal Tool.....	10
5. Burning Guide.....	12
5.1 Install Driver.....	12
5.2 Burn image to eMMC via USB	13
6 EM-IMX8M-MINI Application Guidance.....	16
6.1 Serial Terminal	16
6.2 MIPI LCD and Touch	17
6.3 SD Card.....	18
6.4 USB Host	19
6.5 RTC	19
6.6 Audio Player	20
6.7 Recording	20
6.8 Ethernet.....	21
6.9 CAN	22
6.10 RS485	23
6.11 UART(J14, COM1).....	25

1 EM-IMX8M-MINI Introduction

1.1 Summary

The EM-IMX8M-MINI SBC (single board computer) incorporates SOM-IMX8M-MINI SODIMM module which is based on NXP's energy efficient i.MX8M Mini ARM Cortex A53 and Cortex-M4 processor.

This i.MX8M SBC is tailor made for a wide range of multimedia applications, featuring 2GB LPDDR4, 8GB eMMC, 2 x USB 2.0, powerful network connectivity options including 4G, WiFi and Bluetooth. Robust multimedia features including 1080P60 video HEVC/H265/H264/VP9 decode with HDR, 2D/3D graphics acceleration, 16 audio channels (32bits), MIPI-DSI, and 1080p encoder and decoder. EM-IMX8M-MINI is ideal for Advanced graphics, machine vision, and other multimedia applications.

1.2 Processor Features

CPU

- 4x Cortex-A53 core platforms up to 1.8GHz per core
- 32KB L1-I Cache/ 32 kB L1-D Cache
- 512 kB L2 Cache
- 1x Arm Cortex-M4 core up to 400MHz
- 16 kB L1-I Cache/ 16 kB L2-D Cache

GPU

- 3D GPU (1x shader, OpenGL® ES 2.0)
- 2D GPU

Video Engine

- 1080p60 VP9 Profile 0, 2 (10-bit) decoder, HEVC/H.265 decoder, AVC/H.264 Baseline, Main, High decoder, VP8 decoder
- 1080p60 AVC/H.264 encoder, VP8 encoder

Camera

- 1x MIPI CSI (4-lane) with PHY

Display

- Content can be display on 4-lane MIPI DSI display.

Audio

- 5x SAI (12Tx + 16Rx external I2S lanes), 8ch PDM input

Memory

- The external memory interfaces supported on this chip include:
 - 16/32-bit DRAM Interface:
 - LPDDR4-3000
 - DDR4-2400
 - DDR3L-1600

1.3 EM-IMX8M-MINI specifications

Processor – i.MX 8MQuad, 4x ARM Cortex-A53 @1.8 GHz, 1 MB L2 cache, ARM Cortex-M4 @400 MHz

GPU – 2D/3D acceleration, support OpenGL ES 1.1, 2.0, OpenVG 1.1

RAM – 2GB LPDDR4

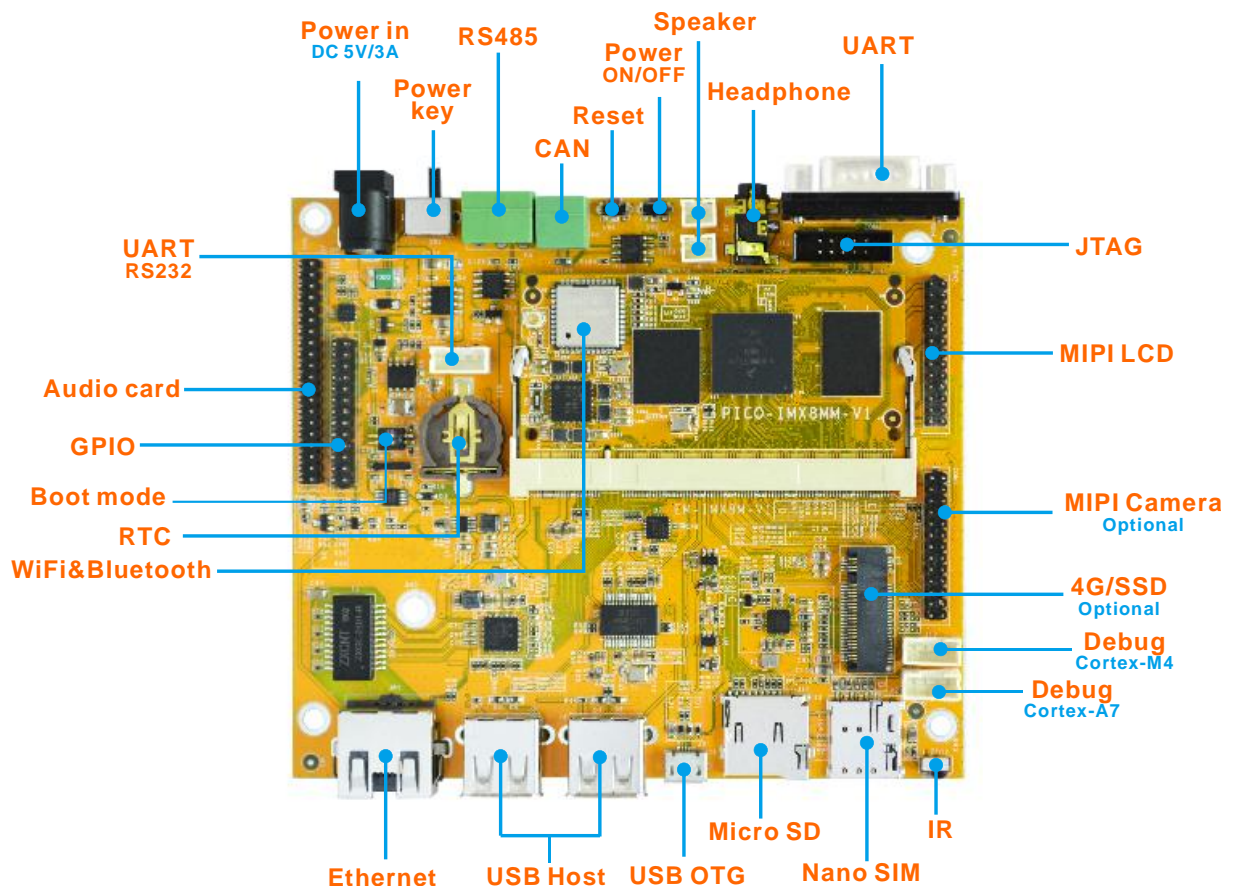
Storage – 8GB

Interfaces – Ethernet, 4x UART, IR, 2x USB Host, USB OTG, PCI-E, CAN, RS485, MIPI-LCD, Camera, GPIO, Audio I/O, SD, SIM, WIFI&Bluetooth, etc.

Operating system – Linux4.14.98

Application – Industrial control, communications and measurement, etc.

Dimension – Baseboard - 102.3mm x 118.6mm; CPU board - 67.6mm x 34.3 mm



2. Compiler Environment

2.1 Vmware10.0+ubuntu16.04

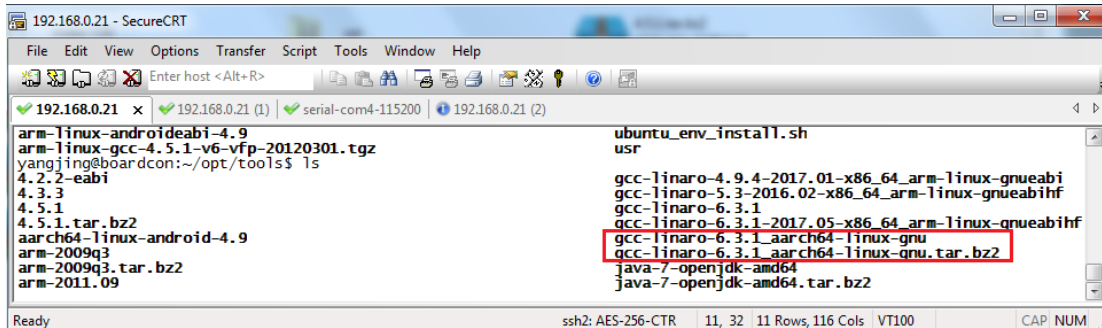
Install Vmware10.0 in windows OS, and then install ubuntu16.04 in VMware to compile. Please refer to the official website <http://www.ubuntu.com/> to download and install Ubuntu system.

Note: Linux should be compiled by ubuntu 64-bit OS.

2.2 Install Tools

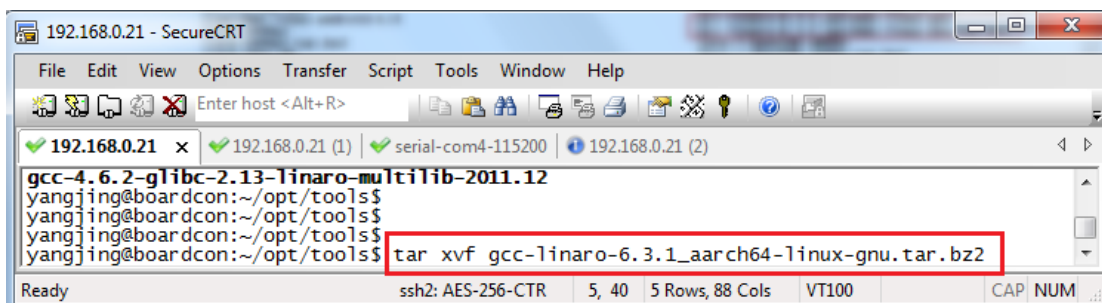
Copy the file: [Source/gcc-linaro-6.3.1_aarch64-linux-gnu.tar.bz2](#) to Ubuntu system, and unzip it:

```
$ tar xvf gcc-linaro-6.3.1_aarch64-linux-gnu.tar.bz2
```



```

arm-linux-androideabi-4.9
arm-linux-gcc-4.5.1-v6-vfp-20120301.tgz
yangjing@boardcon:~/opt/tools$ ls
4.2.2-eabi
4.3.3
4.5.1
4.5.1.tar.bz2
aarch64-linux-android-4.9
arm-2009q3
arm-2009q3.tar.bz2
arm-2011.09
ubuntu_env_install.sh
usr
gcc-linaro-4.9.4-2017.01-x86_64_arm-linux-gnueabi
gcc-linaro-5.3-2016.02-x86_64_arm-linux-gnueabihf
gcc-linaro-6.3.1
gcc-linaro-6.3.1-2017.05-x86_64_arm-linux-gnueabihf
gcc-linaro-6.3.1_aarch64-linux-gnu
gcc-linaro-6.3.1_aarch64-linux-gnu.tar.bz2
java-7-openjdk-amd64
java-7-openjdk-amd64.tar.bz2
  
```

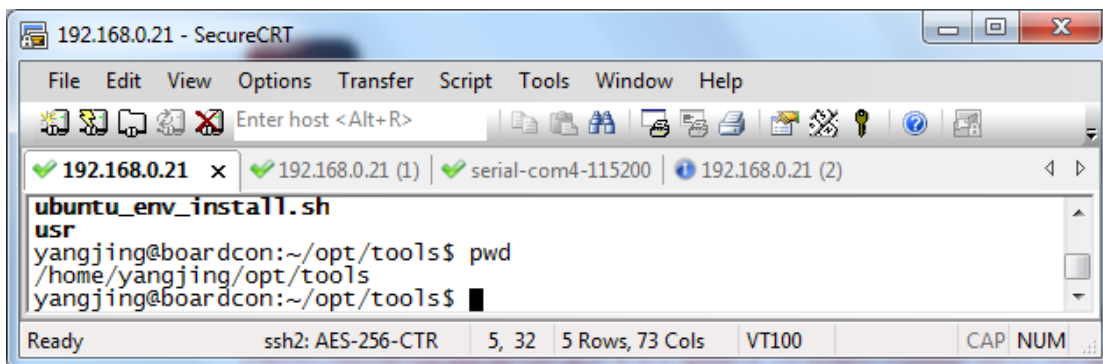


```

gcc-4.6.2-glibc-2.13-linaro-multilib-2011.12
yangjing@boardcon:~/opt/tools$
yangjing@boardcon:~/opt/tools$
yangjing@boardcon:~/opt/tools$ tar xvf gcc-linaro-6.3.1_aarch64-linux-gnu.tar.bz2
yangjing@boardcon:~/opt/tools$
  
```

After unzip finish can get the files [gcc-linaro-6.3.1_aarch64-linux-gnu](#). Execute the follow command to view current directory:

```
$ pwd
```



```

ubuntu_env_install.sh
usr
yangjing@boardcon:~/opt/tools$ pwd
/home/yangjing/opt/tools
yangjing@boardcon:~/opt/tools$
  
```

Execute the follow command to set the compiler effective.

```
$ vi ~/.bashrc
```

Then add the follow content in the last line.

```
export PATH=/home/yangjing/opt/tools/gcc-linaro-6.3.1_aarch64-linux-gnu/bin:$PATH
```

Note

The path [/home/yangjing/opt/tools](#) is user's Ubuntu system path of storage [gcc-linaro-6.3.1_aarch64-linux-gnu](#).


```

192.168.0.21 - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
192.168.0.21 x 192.168.0.21 (1) serial-com4-115200 192.168.0.21 (2)
jdk1.7.0_80.tar.bz2
make-3.81
ubuntu_env_install.sh
usr
yangjing@boardcon:~/opt/tools$ vi ~/.bashrc
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

yangjing@boardcon:~/opt/tools$ source ~/.bashrc

* keychain 2.8.1 ~ http://www.funtoo.org
  
```

```

192.168.0.21 - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
192.168.0.21 x 192.168.0.21 (1) serial-com4-115200 192.168.0.21 (2)
for imx8m-mini linux
export PATH=/home/yangjing/opt/tools/gcc-linaro-6.3.1_aarch64-linux-gnu/bin:$PATH
173,1 81%
Ready ssh2: AES-256-CTR 2, 1 5 Rows, 101 Cols VT100 CAP NUM
  
```

Save and close the script. Execute the follow command to set the compiler effective.

`$ source ~/.bashrc`

Execute the command to view currently valid compiler.

`$ aarch64-linux-gnu-gcc -v`

```

192.168.0.21 - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
192.168.0.21 x 192.168.0.21 (1) serial-com4-115200 192.168.0.21 (2)
yangjing@boardcon:~/opt/tools$
yangjing@boardcon:~/opt/tools$
yangjing@boardcon:~/opt/tools$ aarch64-linux-gnu-gcc -v
Using built-in specs.
COLLECT_GCC=aarch64-linux-gnu-gcc
COLLECT_LTO_WRAPPER=/home/yangjing/opt/tools/gcc-linaro-6.3.1_aarch64-linux-gnu/bin/../libexec/gcc/aarch64-linux-gnu/6.3.1/lto-wrapper
Target: aarch64-linux-gnu
Configured with: '/home/tcwg-buildslave/workspace/tcwg-make-release/builder_arch/amd64/label/tcwg-x86_64-build/target/aarch64-linux-gnu/snapshots/gcc.git-linaro-6.3-2017.05/configure' SHELL=/bin/ash --with-mpc=/home/tcwg-buildslave/workspace/tcwg-make-release/builder_arch/amd64/label/tcwg-x86_64-build/target/aarch64-linux-gnu/_build/builds/destdir/x86_64-unknown-linux-gnu --with-mpfr=/home/tcwg-buildslave/workspace/tcwg-make-release/builder_arch/amd64/label/tcwg-x86_64-build/target/aarch64-linux-gnu/_build/builds/destdir/x86_64-unknown-linux-gnu --with-gmp=/home/tcwg-buildslave/workspace/tcwg-make-release/builder_arch/amd64/label/tcwg-x86_64-build/target/aarch64-linux-gnu/_build/builds/destdir/x86_64-unknown-linux-gnu --with-gnu-as --with-gnu-ld --disable-libmudflap --enable-lto --enable-shared --without-included-gettext --enable-nls --disable-sjlj-exceptions --enable-gnu-unique-object --enable-linker-build-id --disable-libstdcxx-pch --enable-c99 --enable-clocale-gnu --enable-libstdcxx-debug --enable-long-long --with-cloog=no --with-pp1=no --with-isl=no --disable-multilib --enable-fix-cortex-a53-835769 --enable-fix-cortex-a53-843419 --with-arch=armv8-a --enable-threads=posix --enable-multiarch --enable-libstdcxx-time=yes --enable-gnu-indirect-function --with-build-sysroot=/home/tcwg-buildslave/workspace/tcwg-make-release/builder_arch/amd64/label/tcwg-x86_64-build/target/aarch64-linux-gnu/_build/builds/destdir/x86_64-unknown-linux-gnu/aarch64-linux-gnu/libc --enable-checking=release --disable-bootstrap --enable-languages=c,c++,fortran,lto --build=x86_64-unknown-linux-gnu --host=x86_64-unknown-linux-gnu --target=aarch64-linux-gnu --prefix=/home/tcwg-buildslave/workspace/tcwg-make-release/builder_arch/amd64/label/tcwg-x86_64-build/target/aarch64-linux-gnu/_build/builds/destdir/x86_64-unknown-linux-gnu
Thread model: posix
gcc version 6.3.1 20170404 (Linaro GCC 6.3-2017.05)
yangjing@boardcon:~/opt/tools$
Ready ssh2: AES-256-CTR 33, 32 33 Rows, 98 Cols VT100 CAP NUM
  
```


3. Compile the Source

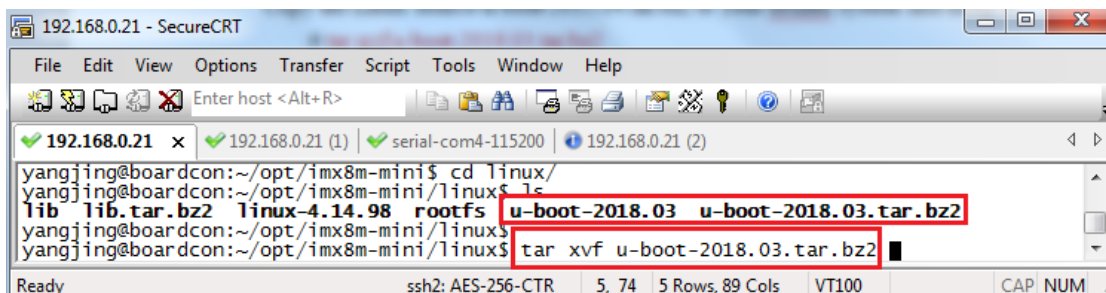
Source	Path
Compiler	Source\gcc-linaro-6.3.1_aarch64-linux-gnu.tar.bz2
Uboot	Source\u-boot-2018.03.tar.bz2
Kernel	Source\linux-4.14.98.tar.bz2
Rootfs	Source\rootfs.tar.bz2

3.1 Compile Uboot

Step 1, unzip the source.

Copy [Source\u-boot-2018.03.tar.bz2](#) to ubuntu system and unzip.

```
$ tar xvf u-boot-2018.03.tar.bz2
```



```

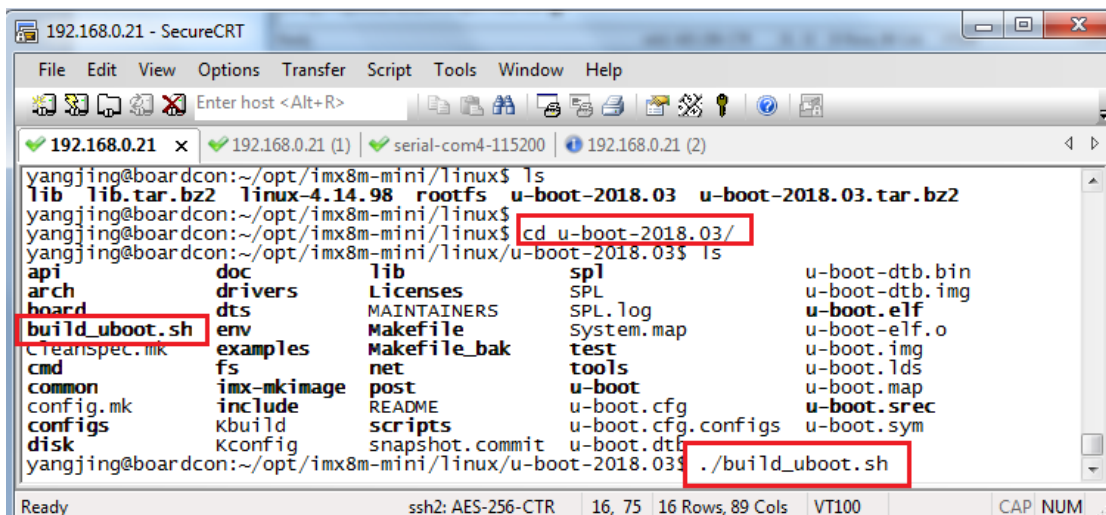
192.168.0.21 - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
192.168.0.21 x 192.168.0.21 (1) serial-com4-115200 192.168.0.21 (2)
yangjing@boardcon:~/opt/imx8m-mini$ cd linux/
yangjing@boardcon:~/opt/imx8m-mini/linux$ ls
lib lib.tar.bz2 linux-4.14.98 rootfs u-boot-2018.03 u-boot-2018.03.tar.bz2
yangjing@boardcon:~/opt/imx8m-mini/linux$ tar xvf u-boot-2018.03.tar.bz2

```

Step 2, compile

```
$ cd u-boot-2018.03/
```

```
$ ./build_uboot.sh
```



```

192.168.0.21 - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
192.168.0.21 x 192.168.0.21 (1) serial-com4-115200 192.168.0.21 (2)
yangjing@boardcon:~/opt/imx8m-mini/linux$ ls
lib lib.tar.bz2 linux-4.14.98 rootfs u-boot-2018.03 u-boot-2018.03.tar.bz2
yangjing@boardcon:~/opt/imx8m-mini/linux$ cd u-boot-2018.03/
yangjing@boardcon:~/opt/imx8m-mini/linux/u-boot-2018.03$ ls
api doc lib spl u-boot-dtb.bin
arch drivers Licenses SPL u-boot-dtb.img
board dts MAINTAINERS SPL.log u-boot.elf
build_uboot.sh env Makefile System.map u-boot-elf.o
Cleanspec.mk examples Makefile_bak test u-boot.img
cmd fs net tools u-boot.ld5
common imx-mkimage post u-boot u-boot.map
config.mk include README u-boot.cfg u-boot.srec
configs Kbuild scripts u-boot.cfg.configs u-boot.sym
disk Kconfig snapshot.commit u-boot.dtb
yangjing@boardcon:~/opt/imx8m-mini/linux/u-boot-2018.03$ ./build_uboot.sh

```

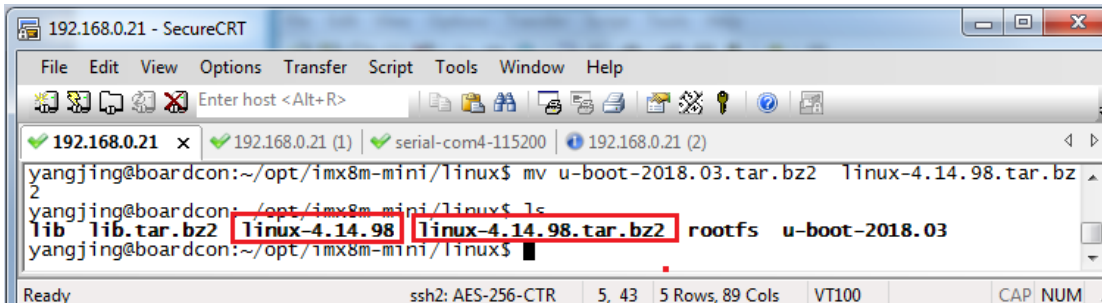
After compiling, `imx-boot-imx8mmevk-sd.bin-flash_evk` are generated in the current directory.

3.2 Compile Kernel

Step 1, unzip the source.

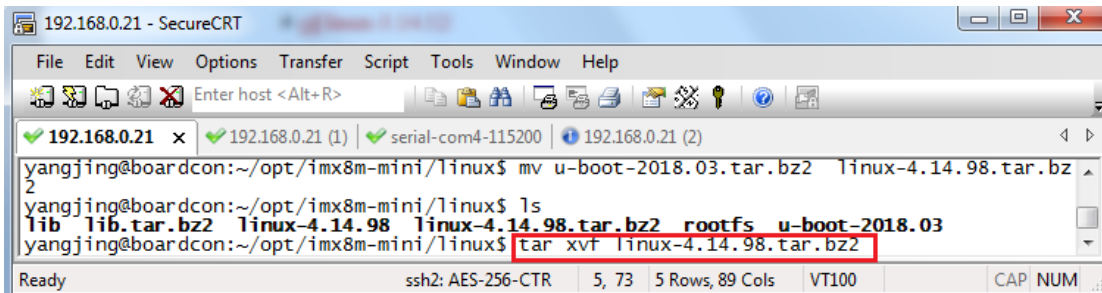
Copy [Source\linux-4.14.98.tar.bz2](#) to ubuntu system and unzip it.

\$ tar xvf linux-4.14.98.tar.bz2



```

yangjing@boardcon:~/opt/imx8m-mini/linux$ mv u-boot-2018.03.tar.bz2 linux-4.14.98.tar.bz2
yangjing@boardcon:~/opt/imx8m-mini/linux$ ls
lib lib.tar.bz2 linux-4.14.98 linux-4.14.98.tar.bz2 rootfs u-boot-2018.03
yangjing@boardcon:~/opt/imx8m-mini/linux$
  
```



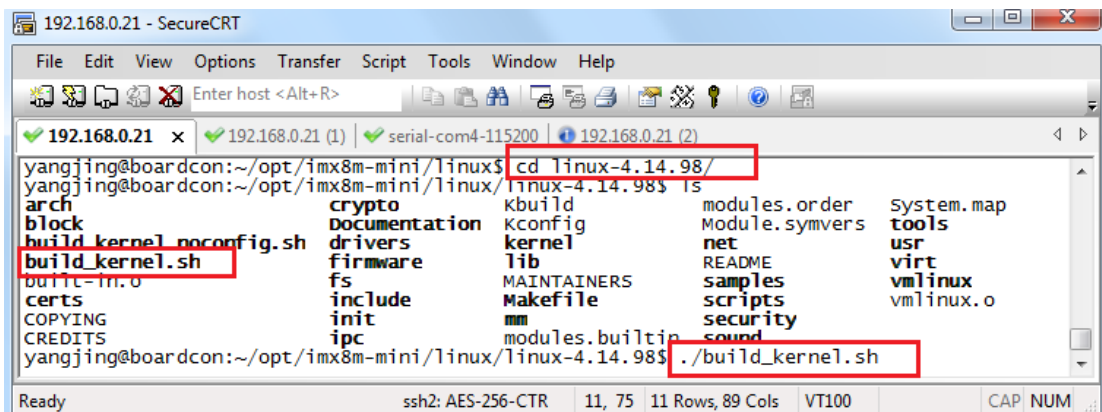
```

yangjing@boardcon:~/opt/imx8m-mini/linux$ mv u-boot-2018.03.tar.bz2 linux-4.14.98.tar.bz2
yangjing@boardcon:~/opt/imx8m-mini/linux$ ls
lib lib.tar.bz2 linux-4.14.98 linux-4.14.98.tar.bz2 rootfs u-boot-2018.03
yangjing@boardcon:~/opt/imx8m-mini/linux$ tar xvf linux-4.14.98.tar.bz2
  
```

Step 2, compile

\$ cd linux-4.14.98/

\$. /build_kernel.sh

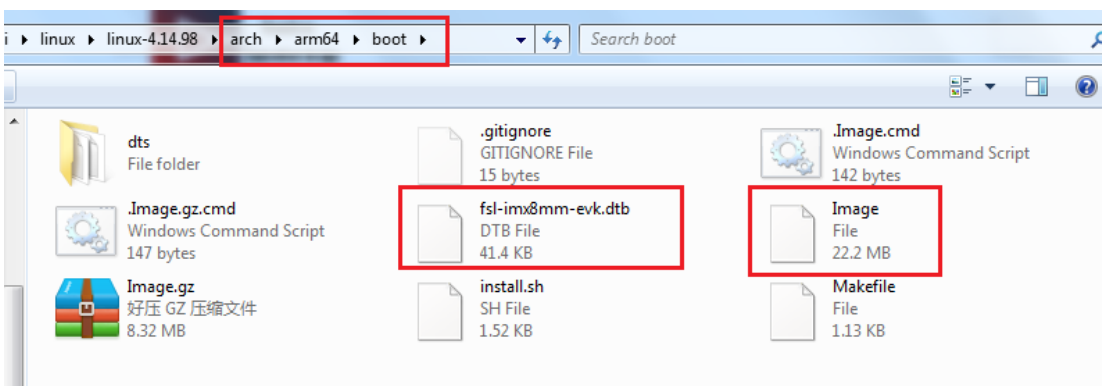


```

yangjing@boardcon:~/opt/imx8m-mini/linux$ cd linux-4.14.98/
yangjing@boardcon:~/opt/imx8m-mini/linux/linux-4.14.98$ ls
arch          crypto        kbuild        modules.order  System.map
block        Documentation Kconfig       Module.symvers tools
build_kernel_noconfig.sh  drivers       kernel         net            usr
build_kernel.sh  firmware     lib           README         virt
bootctl-init  fs           MAINTAINERS  samples        vmlinux
certs         include     Makefile      security       vmlinux.o
COPYING      ipc         mm            sound
CREDITS      init        modules.builtin
yangjing@boardcon:~/opt/imx8m-mini/linux/linux-4.14.98$ ./build_kernel.sh
  
```

After compiling, **Image** and **fsl-imx8mm-evk.dtb** are generated in the directory

`linux-4.14.98/arch/arm64/boot`



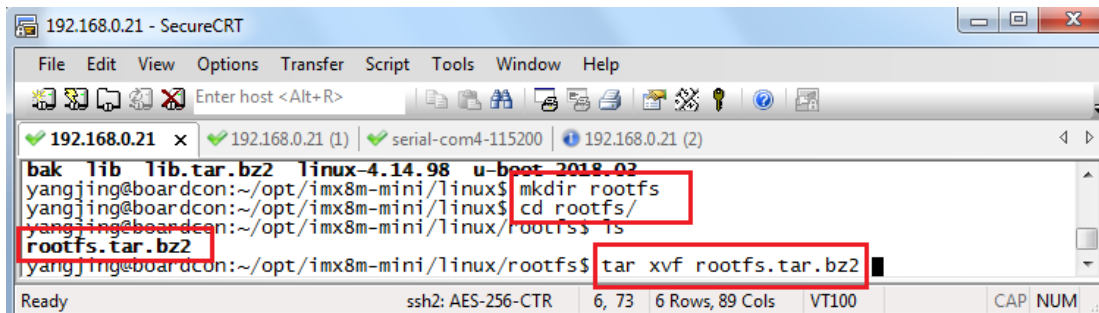
3.3 Compile rootfs

The root file system not need to compile, only compression or decompression. If want to add files to the rootfs file system, just copy [Source\rootfs.tar.bz2](#) to ubuntu system and unzip it:

```
$ mkdir rootfs
```

```
$ cd rootfs (Copy Source\rootfs.tar.bz2 to currently directory)
```

```
$tar xvf rootfs.tar.bz2
```

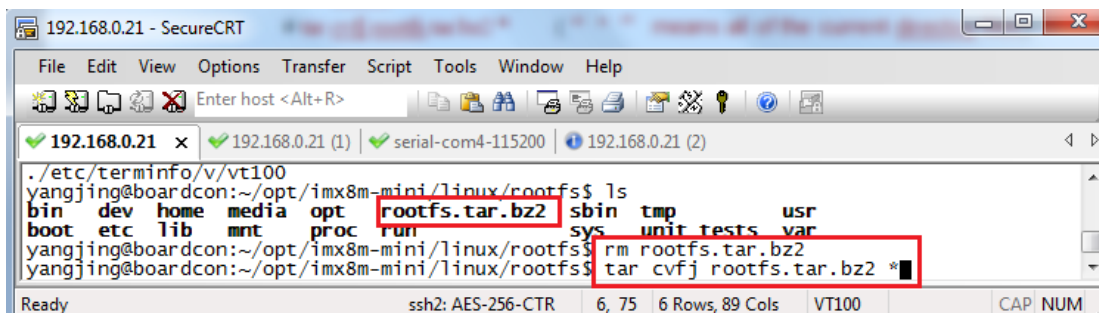


```

192.168.0.21 - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
192.168.0.21 x 192.168.0.21 (1) serial-com4-115200 192.168.0.21 (2)
bak lib lib.tar.bz2 linux-4.14.98 u-boot 2018.02
yangjing@boardcon:~/opt/imx8m-mini/linux$ mkdir rootfs
yangjing@boardcon:~/opt/imx8m-mini/linux$ cd rootfs/
yangjing@boardcon:~/opt/imx8m-mini/linux/rootfs$ ls
rootfs.tar.bz2
yangjing@boardcon:~/opt/imx8m-mini/linux/rootfs$ tar xvf rootfs.tar.bz2
  
```

```
$ rm rootfs.tar.bz2 (delete the old file system)
```

```
$ tar cvfj rootfs.tar.bz2 * ("*" means all of the current directory files)
```



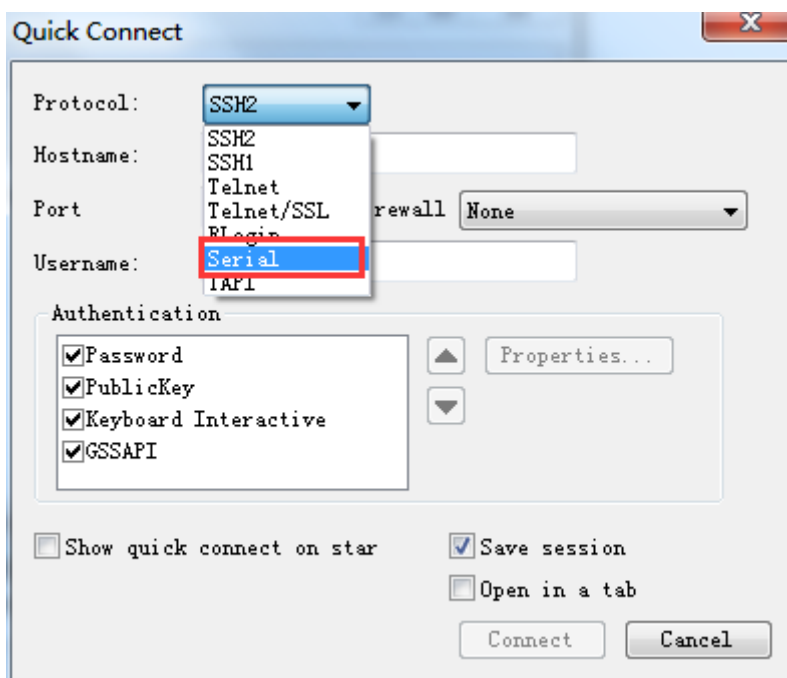
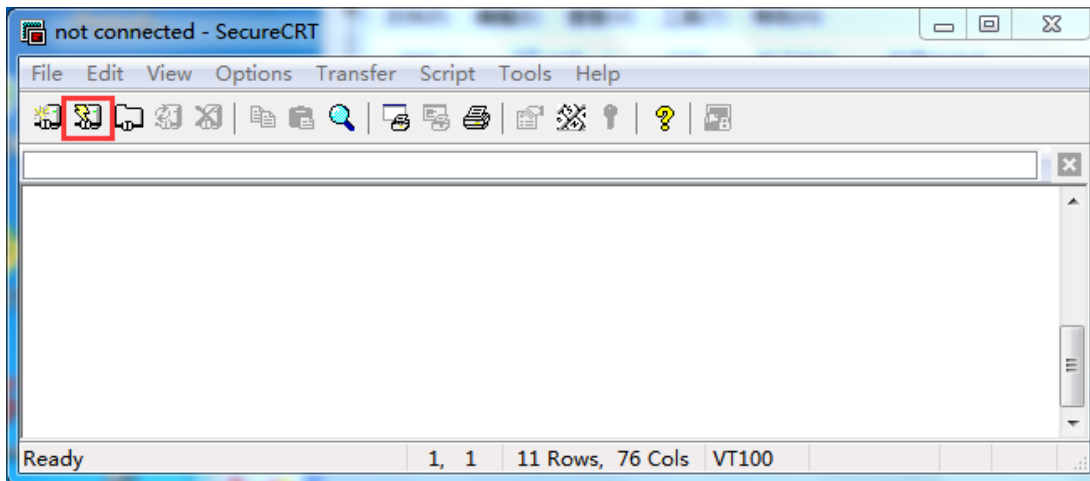
```

192.168.0.21 - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
192.168.0.21 x 192.168.0.21 (1) serial-com4-115200 192.168.0.21 (2)
./etc/terminfo/v/vt100
yangjing@boardcon:~/opt/imx8m-mini/linux/rootfs$ ls
bin dev home media opt rootfs.tar.bz2 sbin tmp usr
boot etc lib mnt proc run sys unit tests var
yangjing@boardcon:~/opt/imx8m-mini/linux/rootfs$ rm rootfs.tar.bz2
yangjing@boardcon:~/opt/imx8m-mini/linux/rootfs$ tar cvfj rootfs.tar.bz2 *
  
```

4 Install Serial Terminal Tool

The serial terminal SecureCRT is used for debugging. It can be used directly after decompression.

Open SecureCRT.exe after copy to PC (path: [tools\windows\SecureCRT.exe](#)), then click the icon **Quick Connect** to config.



Set the parameters as follow:

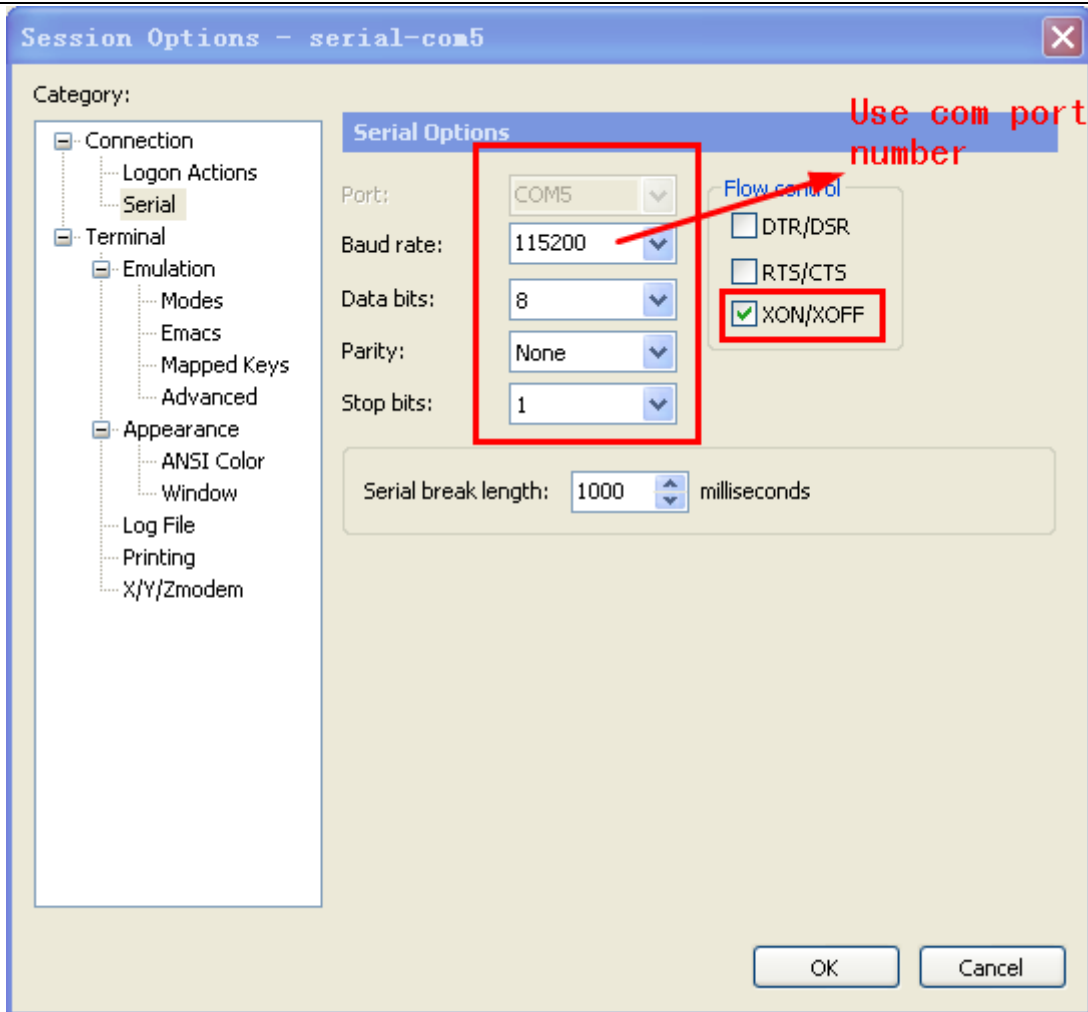
Protocol: Serial

Port: To be specified by user PC

Baud rate: 115200

Please check XON/XOFF but not RTS/CTS

Check Save session



After all, click **connect**

Illusion1: If open more than one serial terminal tools, and they use the same serial port, there will be reported **the port is busy**.

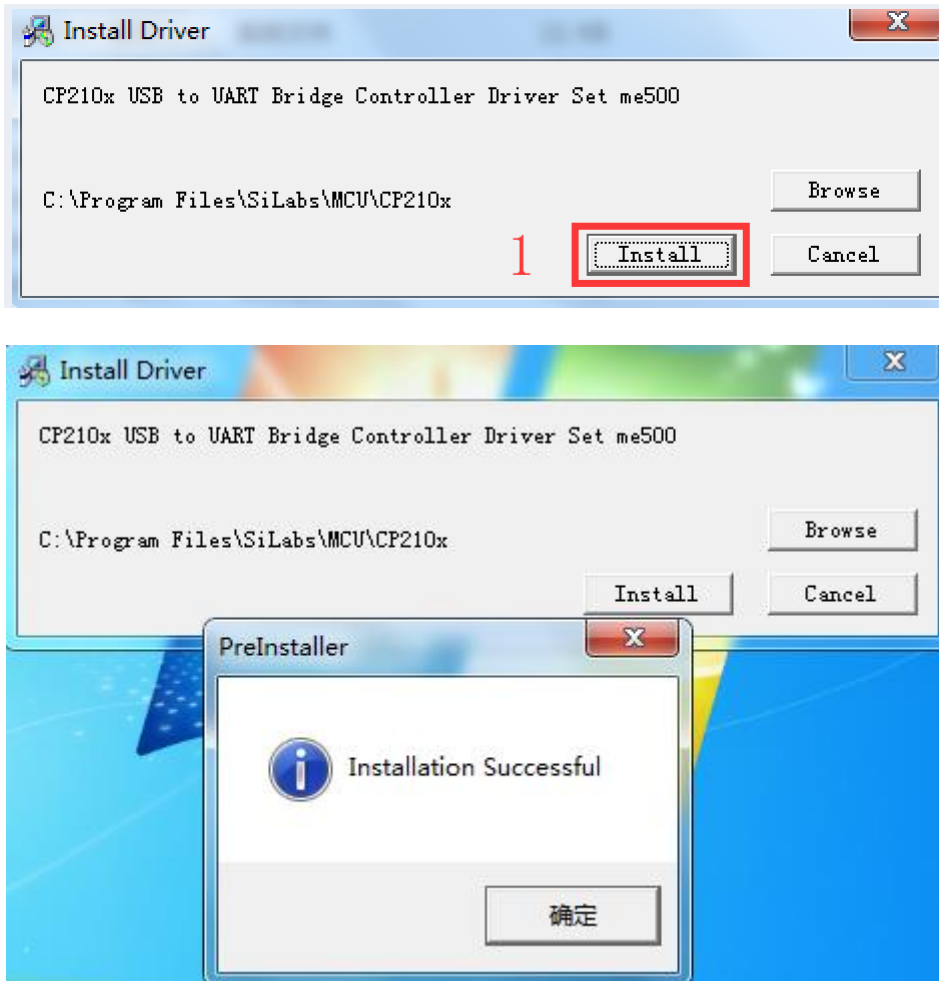
Solution: Turn off the serial tool that unnecessary.

5. Burning Guide

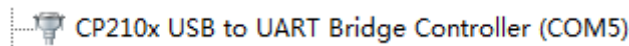
5.1 Install Driver

Install **CP2102 driver**.

Plug the **USB-to-UART cable CP2102** to the PC, unzip **CP2102WIN7.rar** on Windows, then click **preinstaller.exe** to install






Now the device will be listed under **Device Manager** -> **PORTS** with unique serial port assigned



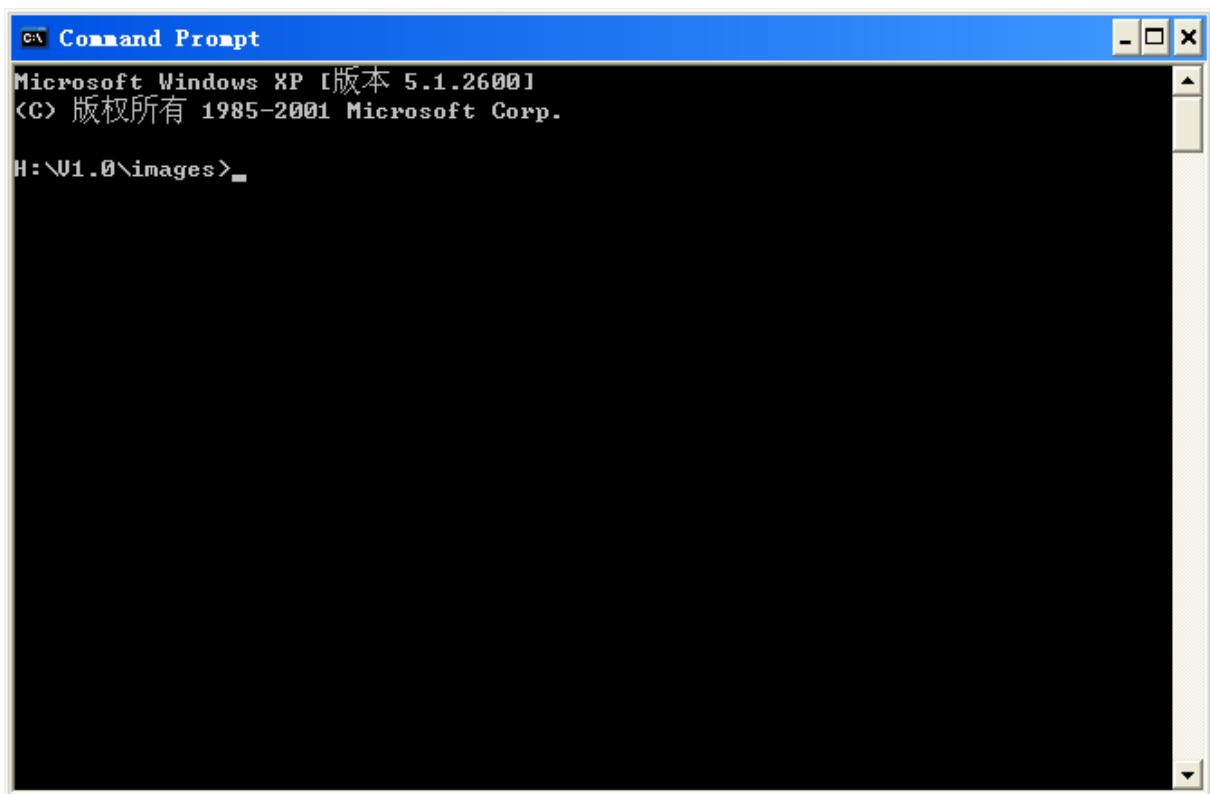
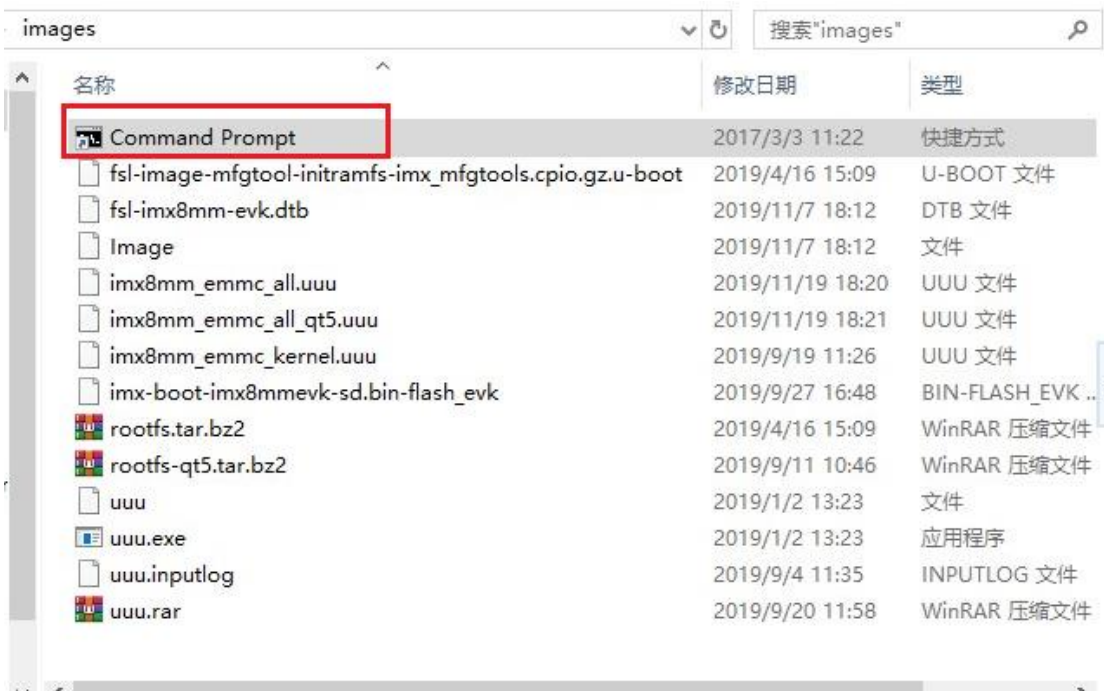
5.2 Burn image to eMMC via USB

Step 1, Set switch to download mode (SW4: OFF OFF).

SW4	1	2	 = ON
Download	OFF	OFF	
eMMC Boot	ON	ON	

Default set **eMMC** as normal start mode.

Step 2, Copy **images** to Windows PC. Open **Command Prompt**.



Step 3, Connect development board to PC with USB OTG cable and serial cable, then power on(5V).

Step 4, Execute follow command in Command Prompt start to download:

`uuu imx8mm_emmc_all.uuu` (download uboot, kernel, rootfs.tar.bz2)

or

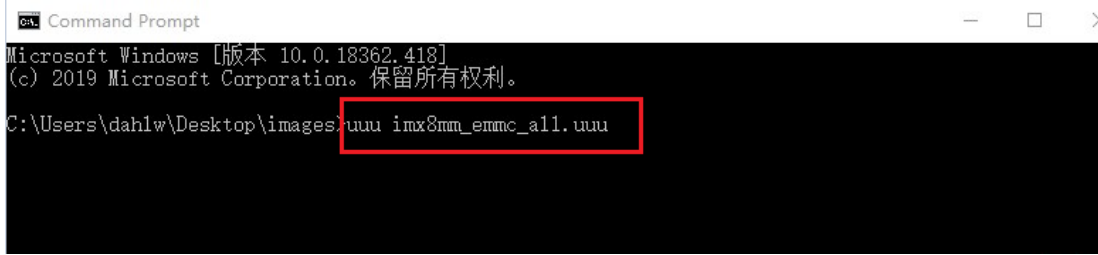
`uuu imx8mm_emmc_all_qt5.uuu` (download uboot, kernel, rootfs-qt5.tar.bz2)

or

[uuu imx8mm_emmc_kernel.uuu](#) (download uboot and kernel)

Note

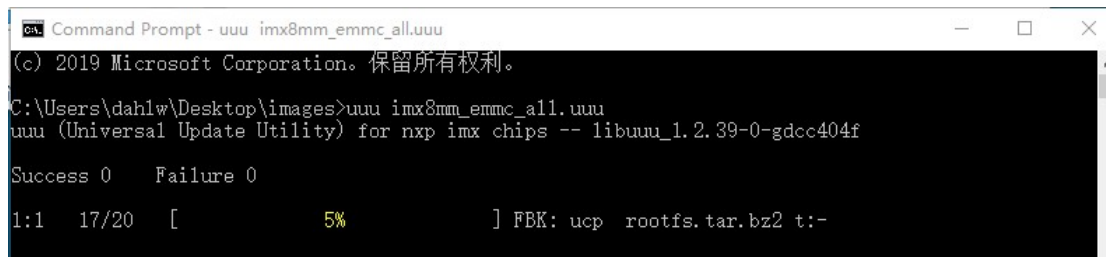
The **uuu** only can be used in **windows 10**. Please download the images in windows 10 system.



```

Command Prompt
Microsoft Windows [版本 10.0.18362.418]
(c) 2019 Microsoft Corporation。保留所有权利。

C:\Users\dahlw\Desktop\images>uuu imx8mm_emmc_all.uuu
  
```



```

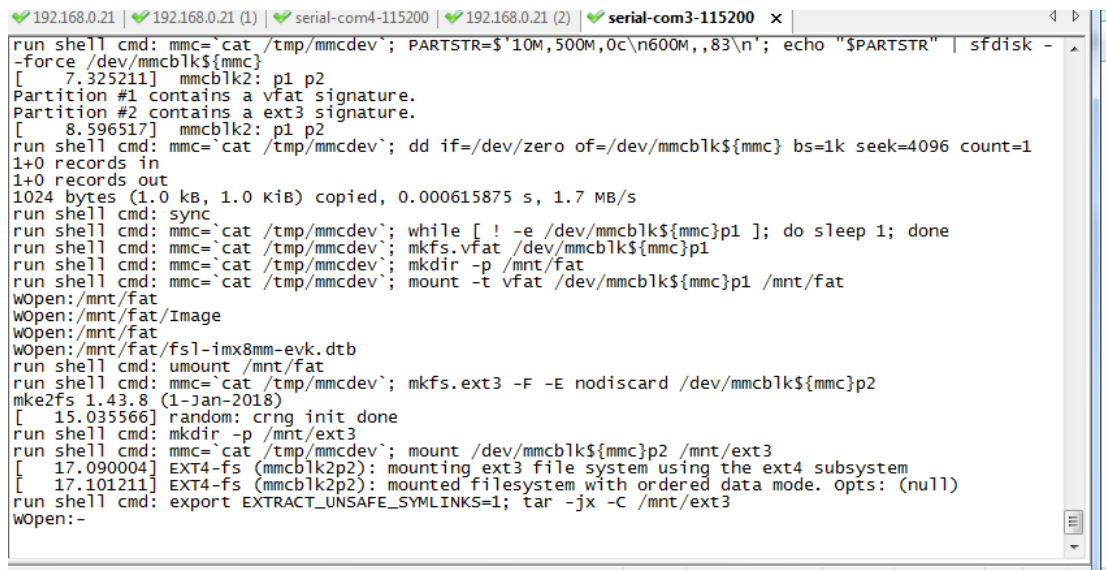
Command Prompt - uuu imx8mm_emmc_all.uuu
(c) 2019 Microsoft Corporation。保留所有权利。

C:\Users\dahlw\Desktop\images>uuu imx8mm_emmc_all.uuu
uuu (Universal Update Utility) for nxp imx chips -- libuuu_1.2.39-0-gdcc404f

Success 0   Failure 0

1:1  17/20  [          5%          ] FBK: ucp rootfs.tar.bz2 t:-
  
```

The SecureCRT will output the download messages.



```

192.168.0.21 | 192.168.0.21 (1) | serial-com4-115200 | 192.168.0.21 (2) | serial-com3-115200 x
run shell cmd: mmc=`cat /tmp/mmcdev`; PARTSTR='$'10M,500M,0C\n600M,,83\n'; echo "$PARTSTR" | sfdisk -
-force /dev/mmcblk${mmc}
[ 7.325211] mmcblk2: p1 p2
Partition #1 contains a vfat signature.
Partition #2 contains a ext3 signature.
[ 8.596517] mmcblk2: p1 p2
run shell cmd: mmc=`cat /tmp/mmcdev`; dd if=/dev/zero of=/dev/mmcblk${mmc} bs=1k seek=4096 count=1
1+0 records in
1+0 records out
1024 bytes (1.0 kB, 1.0 KiB) copied, 0.000615875 s, 1.7 MB/s
run shell cmd: sync
run shell cmd: mmc=`cat /tmp/mmcdev`; while [ ! -e /dev/mmcblk${mmc}p1 ]; do sleep 1; done
run shell cmd: mmc=`cat /tmp/mmcdev`; mkfs.vfat /dev/mmcblk${mmc}p1
run shell cmd: mmc=`cat /tmp/mmcdev`; mkdir -p /mnt/fat
run shell cmd: mmc=`cat /tmp/mmcdev`; mount -t vfat /dev/mmcblk${mmc}p1 /mnt/fat
wOpen:/mnt/fat
wOpen:/mnt/fat/Image
wOpen:/mnt/fat
wOpen:/mnt/fat/fs1-imx8mm-evk.dtb
run shell cmd: umount /mnt/fat
run shell cmd: mmc=`cat /tmp/mmcdev`; mkfs.ext3 -F -E nodiscard /dev/mmcblk${mmc}p2
mke2fs 1.43.8 (1-Jan-2018)
[ 15.035566] random: crng init done
run shell cmd: mkdir -p /mnt/ext3
run shell cmd: mmc=`cat /tmp/mmcdev`; mount /dev/mmcblk${mmc}p2 /mnt/ext3
[ 17.090004] EXT4-fs (mmcblk2p2): mounting ext3 file system using the ext4 subsystem
[ 17.101211] EXT4-fs (mmcblk2p2): mounted filesystem with ordered data mode. opts: (null)
run shell cmd: export EXTRACT_UNSAFE_SYMLINKS=1; tar -jx -C /mnt/ext3
wOpen:-
  
```

Download completed.



```

选择Command Prompt
C:\Users\dahlw\Desktop\images>
C:\Users\dahlw\Desktop\images>
C:\Users\dahlw\Desktop\images>uuu imx8mm_emmc_all.uuu
uuu (Universal Update Utility) for nxp imx chips -- libuuu_1.2.39-0-gdcc404f

Success 1   Failure 0

1:1  20/20  [Done          ] FBK: DONE

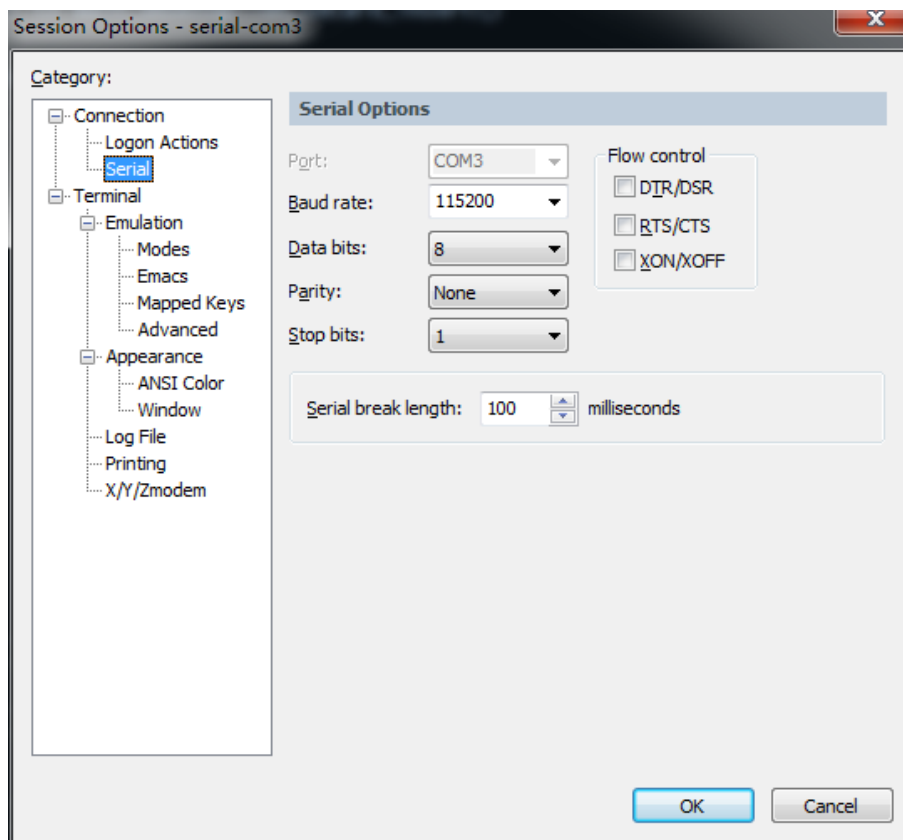
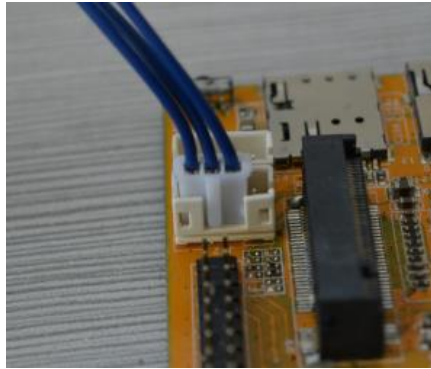
C:\Users\dahlw\Desktop\images>
  
```

After finish, set SW4 to ON ON, then repower the board to startup.

6 EM-IMX8M-MINI Application Guidance

6.1 Serial Terminal

Connect the board J12 and PC with USB Serial cable, then power on, the terminal will output startup information.



```

192.168.0.21 | 192.168.0.21 (1) | serial-com4-115200 | 192.168.0.21 (2) | serial-com3-115200 x
lo ethernet found.
astboot: Normal
ormal Boot
it any key to stop autoboot: 0
witch to partitions #0, OK
mc1(part 0) is current device
* Unable to read file boot.scr **
3345664 bytes read in 277 ms (80.4 MiB/s)
ooting from mmc ...
2462 bytes read in 9 ms (4.5 MiB/s)
# Flattened Device Tree blob at 43000000
Booting using the fdt blob at 0x43000000
Using Device Tree in place at 0000000043000000, end 000000004300d5dd

Starting kernel ...

0.000000] Booting Linux on physical CPU 0x0
0.000000] Linux version 4.14.98 (yangjing@boardcon) (gcc version 6.3.1 20170404 (Linaro GCC 6.3
2017.05)) #50 SMP PREEMPT Thu Nov 7 18:04:27 CST 2019
0.000000] Boot CPU: AArch64 Processor [410fd034]
0.000000] Machine model: FSL i.MX8MM EVK board
0.000000] earlycon: ec_imx6q0 at MMIO 0x0000000030890000 (options '115200')
0.000000] bootconsole [ec_imx6q0] enabled
0.000000] efi: Getting EFI parameters from FDT:
0.000000] efi: UEFI not found.
0.000000] Reserved memory: created CMA memory pool at 0x0000000078000000, size 640 MiB
0.000000] OF: reserved mem: initialized node linux,cma, compatible id shared-dma-pool
0.000000] NUMA: No NUMA configuration found
0.000000] NUMA: Faking a node at [mem 0x0000000000000000-0x00000000bfffffff]
0.000000] NUMA: NODE_DATA [mem 0xbdfbe700-0xbdfc04ff]

Ready                               Serial: COM3, 115200  30, 79  30 Rows, 100 Cols  VT100  CAP NUM

```

Input "root" to login the system.

```

serial-com3-115200 - SecureCRT
File Edit View Options Transfer Script Tools Window Help
192.168.0.21 | 192.168.0.21 (1) | serial-com4-115200 | 192.168.0.21 (2) | serial-com3-115200 x

[ OK ] Starting Terminate Psplash Boot Screen...
[ OK ] Started Permit User Sessions.
[ OK ] Started /etc/rc.local compatibility.
[ OK ] Started Terminate Psplash Boot Screen.
[ OK ] Started Hostname Service.
[ OK ] Started Kernel Logging Service.
[ OK ] Started Getty on tty1.
[ OK ] Started Serial Getty on ttyMX1.
[ OK ] Reached target Login Prompts.
Starting Weston wayland Compositor (on tty7)...
[ OK ] Started Weston wayland Compositor (on tty7).
[ OK ] Reached target Multi-User System.
Starting Update UTMP about System Runlevel changes...
[ OK ] Created slice User slice of root.
[ OK ] Started Update UTMP about System Runlevel changes.
[ OK ] Started Session c1 of user root.
Starting User Manager for UID 0...
[ 7.396111] audit: type=1006 audit(1550694262.708:2): pid=3685 uid=0 old-auid=4294967295 auid=0 t
ty=(none) old-ses=4294967295 ses=1 res=1
[ OK ] Started User Manager for UID 0.

NXP i.MX Release Distro 4.14-sumo imx8mmevk ttyMX1

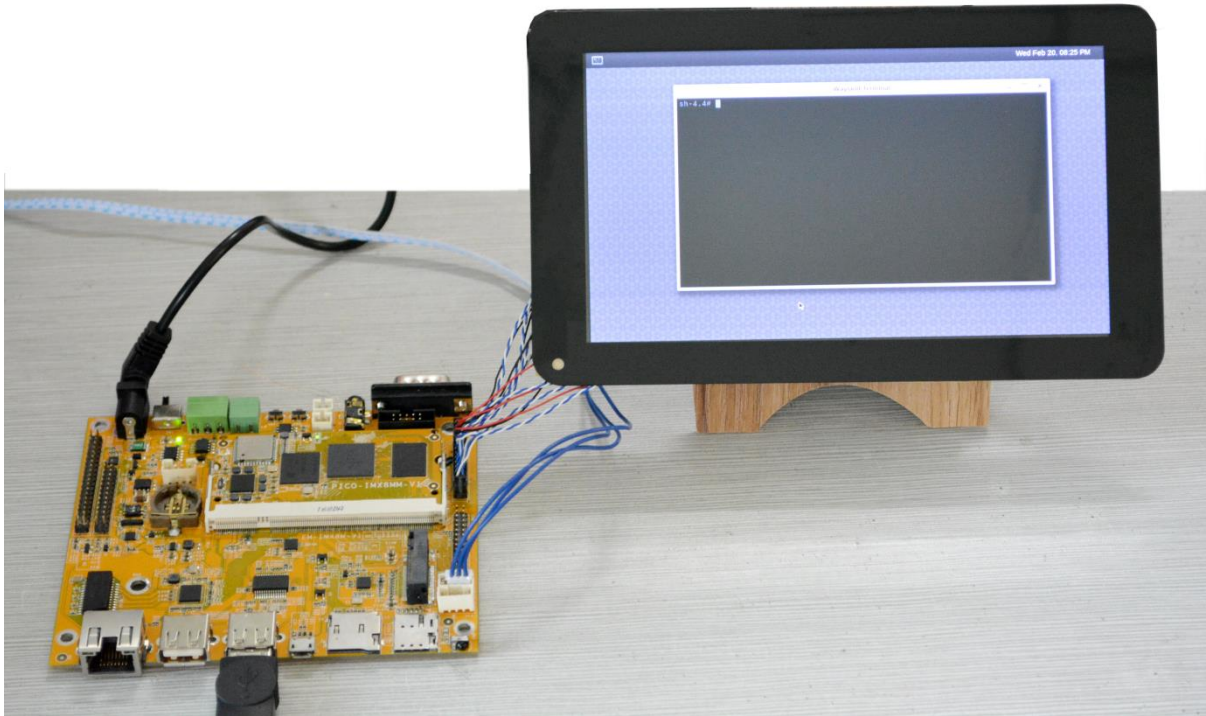
imx8mmevk login: root
Last login: wed Feb 20 20:24:22 UTC 2019 on tty7
root@imx8mmevk:~# cd /
root@imx8mmevk:~# ls
bin  data  etc  lib          media  opt  run  sys      tmp  usr
boot dev home lost+found mnt     proc sbin test.wav unit_tests var
root@imx8mmevk:~#

Ready                               Serial: COM3, 115200  30, 19  30 Rows, 100 Cols  VT100  CAP NUM

```

6.2 MIPI LCD and Touch

Connect MIPI LCD to the board and power on, the QT UI will be displayed.



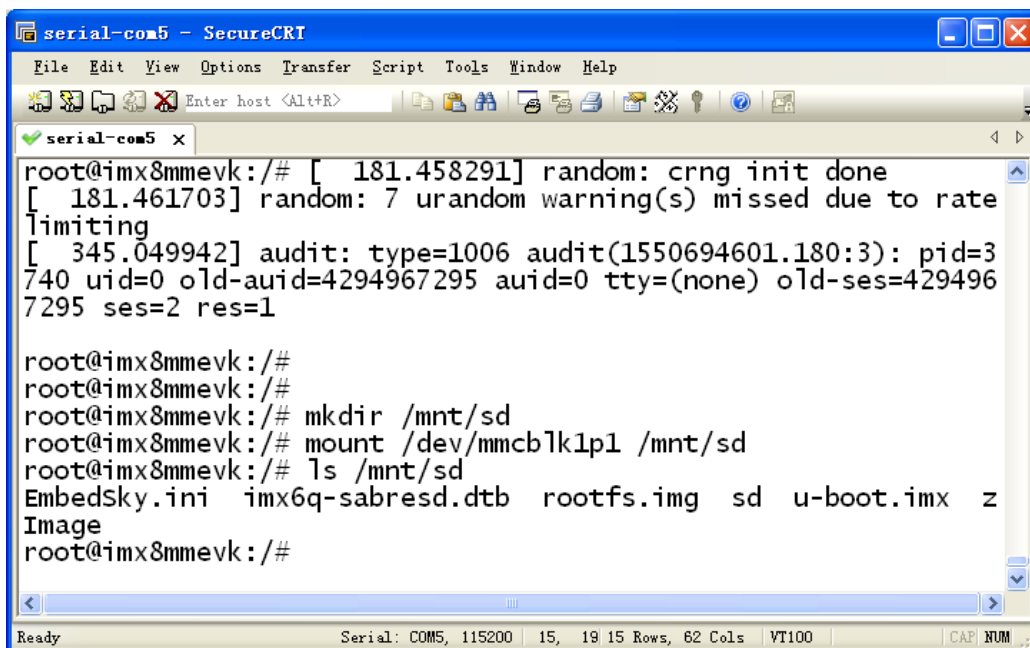
6.3 SD Card

Execute the follow command to mount the SD card and list directory contents of files and directories.

```
# mkdir /mnt/sd
```

```
# mount /dev/mmcblk1p1 /mnt/sd
```

```
# ls /mnt/sd
```



```
serial-com5 - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
serial-com5 x
root@imx8mmevk:/# [ 181.458291] random: crng init done
[ 181.461703] random: 7 urandom warning(s) missed due to rate
limiting
[ 345.049942] audit: type=1006 audit(1550694601.180:3): pid=3
740 uid=0 old-auid=4294967295 auid=0 tty=(none) old-ses=429496
7295 ses=2 res=1

root@imx8mmevk:/#
root@imx8mmevk:/#
root@imx8mmevk:/# mkdir /mnt/sd
root@imx8mmevk:/# mount /dev/mmcblk1p1 /mnt/sd
root@imx8mmevk:/# ls /mnt/sd
EmbedSky.ini imx6q-sabresd.dtb rootfs.img sd u-boot.imx z
Image
root@imx8mmevk:/#

Ready Serial: COM5, 115200 15, 19 15 Rows, 62 Cols VT100 CAP NUM
```

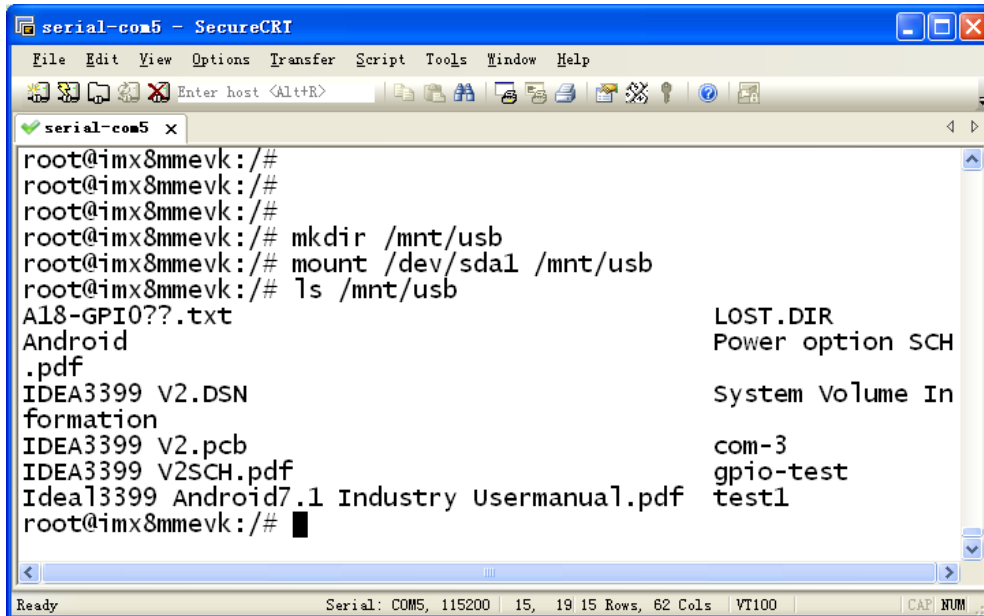
6.4 USB Host

Insert USB device (e.g. U-disk) to USB Host, execute follow command to mount the U-disk.

```
# mkdir /mnt/usb
```

```
# mount /dev/sda1 /mnt/usb
```

```
# ls /mnt/usb
```



```
serial-com5 - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
serial-com5 x
root@imx8mmevk:/#
root@imx8mmevk:/#
root@imx8mmevk:/#
root@imx8mmevk:/# mkdir /mnt/usb
root@imx8mmevk:/# mount /dev/sda1 /mnt/usb
root@imx8mmevk:/# ls /mnt/usb
A18-GPIO???.txt          LOST.DIR
Android                 Power option SCH
.pdf                   System Volume In
IDEA3399 V2.DSN
formation              com-3
IDEA3399 V2.pcb        gpio-test
IDEA3399 V2SCH.pdf     test1
Idea13399 Android7.1 Industry Usermanual.pdf
root@imx8mmevk:/# █
Ready Serial: COM5, 115200 | 15, 19 15 Rows, 62 Cols VT100 CAP NUM
```

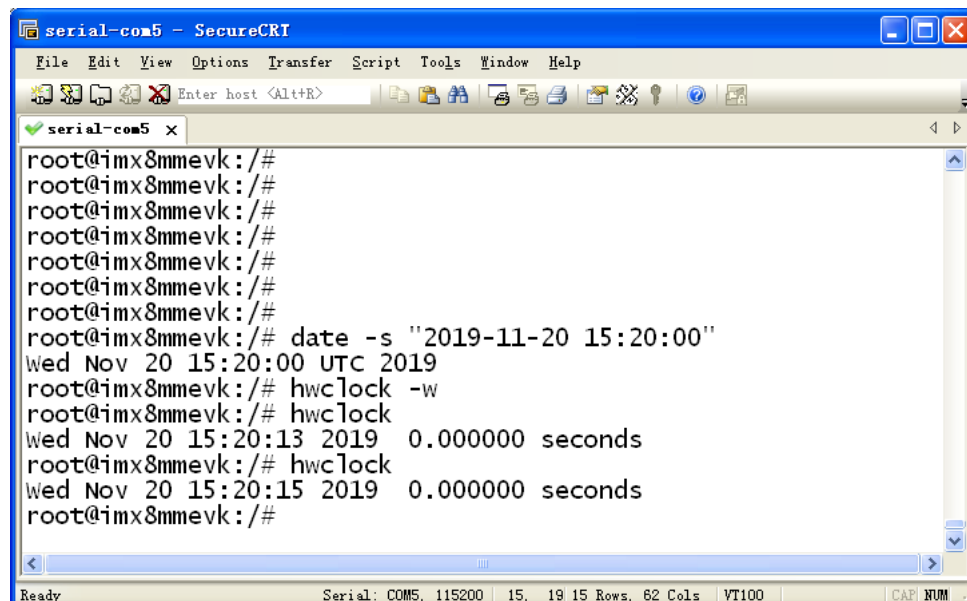
The USB Host also can be used to connect mouse or keyboard.

6.5 RTC

```
# date -s "2019-11-20 15:20:00" (set the system time)
```

```
# hwclock -w
```

```
# hwclock
```

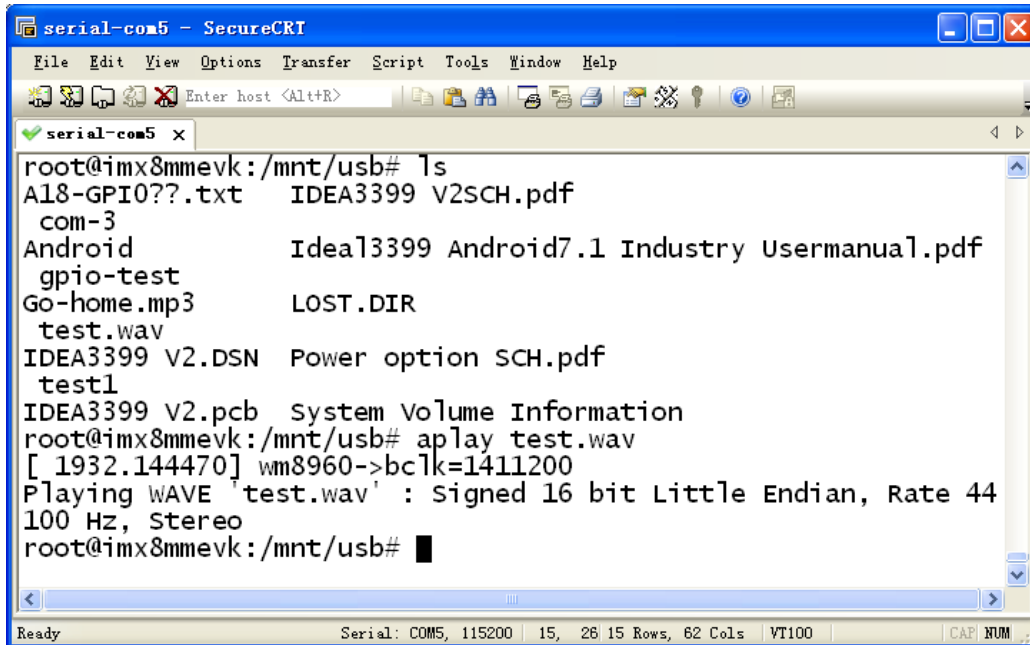


```
serial-com5 - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
serial-com5 x
root@imx8mmevk:/#
root@imx8mmevk:/#
root@imx8mmevk:/#
root@imx8mmevk:/#
root@imx8mmevk:/#
root@imx8mmevk:/#
root@imx8mmevk:/#
root@imx8mmevk:/# date -s "2019-11-20 15:20:00"
Wed Nov 20 15:20:00 UTC 2019
root@imx8mmevk:/# hwclock -w
root@imx8mmevk:/# hwclock
Wed Nov 20 15:20:13 2019 0.000000 seconds
root@imx8mmevk:/# hwclock
Wed Nov 20 15:20:15 2019 0.000000 seconds
root@imx8mmevk:/#
Ready Serial: COM5, 115200 | 15, 19 15 Rows, 62 Cols VT100 CAP NUM
```

6.6 Audio Player

Put **.wav** files to the SD card/U-disk and power on. Execute follow command to test audio.

```
# aplay test.wav
```



```
serial-com5 - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
serial-com5 x
root@imx8mmevk:/mnt/usb# ls
A18-GPI0???.txt  IDEA3399 V2SCH.pdf
com-3
Android          Idea13399 Android7.1 Industry Usermanual.pdf
gpio-test
Go-home.mp3      LOST.DIR
test.wav
IDEA3399 V2.DSN  Power option SCH.pdf
test1
IDEA3399 V2.pcb  System Volume Information
root@imx8mmevk:/mnt/usb# aplay test.wav
[ 1932.144470] wm8960->bclk=1411200
Playing WAVE 'test.wav' : Signed 16 bit Little Endian, Rate 44
100 Hz, Stereo
root@imx8mmevk:/mnt/usb#
```

Speakers (J6, J7) and headphone(J8) output audio sync.

6.7 Recording

Insert the headphone(J8) and execute follow command to record.

```
# arecord -l ( list sound card)
# arecord -f S16_LE -D plughw:0,0 -c 2 test.wav (record)
# aplay test.wav (play)
```



```

serial-com5 - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
serial-com5 x
root@imx8mmevk:/#
root@imx8mmevk:/# arecord -l
**** List of CAPTURE Hardware Devices ****
card 0: wm8960audio [wm8960-audio], device 0: HiFi wm8960-hifi
-0 []
  subdevices: 1/1
  Subdevice #0: subdevice #0
root@imx8mmevk:/# arecord -f S16_LE -D plughw:0,0 -c 2 test.wa
V
Recording WAVE 'test.wav' : Signed 16 bit Little Endian, Rate 8000
Hz, Stereo
^CAborted by signal Interrupt...
root@imx8mmevk:/# aplay test.wav
[ 2786.665666] alloc_contig_range: [780f0, 78100) PFNs busy
[ 2786.673303] wm8960->bc1k=1411200
Playing WAVE 'test.wav' : Signed 16 bit Little Endian, Rate 80
00 Hz, Stereo
root@imx8mmevk:/# aplay test.wav
Ready Serial: COM5, 115200 | 19, 19 19 Rows, 62 Cols | VT100 | CAP NUM
  
```

6.8 Ethernet

Plug in an Ethernet cable (RJ45). Auto obtain IP.

```
# ping 192.168.1.1
```

```
# ping www.boardcon.com
```

```

serial-com5 - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
serial-com5 x
root@imx8mmevk:/#
root@imx8mmevk:/#
root@imx8mmevk:/# ifconfig
eth0      Link encap:Ethernet  HWaddr 72:af:32:4b:dc:c1
          inet addr:192.168.1.101 Bcast:192.168.1.255 Mask:255.255.255.0
          inet6 addr: fe80::70af:32ff:fe4b:dcc1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST DYNAMIC MTU:1500 Metric:1
          RX packets:3278 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5165 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:479305 (468.0 KiB)  TX bytes:465634 (454.7 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING MTU:65536 Metric:1
          RX packets:50 errors:0 dropped:0 overruns:0 frame:0
          TX packets:50 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3884 (3.7 KiB)  TX bytes:3884 (3.7 KiB)

root@imx8mmevk:/# █
Ready Serial: COM5, 115200 | 23, 19 23 Rows, 74 Cols | VT100 | CAP NUM
  
```



```

serial-com5 - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
serial-com5 x
RX packets:3509 errors:0 dropped:0 overruns:0 frame:0
TX packets:5250 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:503167 (491.3 KiB) TX bytes:484081 (472.7 KiB)

lo
  Link encap:Local Loopback
  inet addr:127.0.0.1 Mask:255.0.0.0
  inet6 addr: ::1/128 Scope:Host
  UP LOOPBACK RUNNING MTU:65536 Metric:1
  RX packets:78 errors:0 dropped:0 overruns:0 frame:0
  TX packets:78 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:1000
  RX bytes:5972 (5.8 KiB) TX bytes:5972 (5.8 KiB)

root@imx8mmevk:/# ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=1.31 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=2.09 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=2.12 ms
64 bytes from 192.168.1.1: icmp_seq=4 ttl=64 time=1.30 ms
64 bytes from 192.168.1.1: icmp_seq=5 ttl=64 time=2.10 ms
64 bytes from 192.168.1.1: icmp_seq=6 ttl=64 time=1.31 ms

```

Or execute follow command to set static IP.

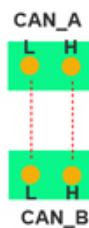
```

# ifconfig eth0 192.168.1.189 up
# route add default gw 192.168.1.1 dev eth0
# ping www.boardcon.com

```

6.9 CAN

Connect CAN ports of Board A and Board B with the test line.



For Board A, execute the follow commands at **Serial terminal A** to set CAN_A as Receiver.

```

# ip link set can0 up type can bitrate 125000 (Bringing CAN0 up and specify bitrate)
# candump can0 (set CAN0 as receive)

```

For Board B, execute the follow commands at **Serial terminal B** to set CAN_B as Transmitter.

```

# ip link set can0 up type can bitrate 125000 (start CAN0)
# cansend can0 123#DEADBEEF (CAN0 send characters 0xDE 0xAD 0xBE 0xEF)

```

The Transmitter and receiver can be converted by execute the command

```

# candump can0 (Receiver)

```

or

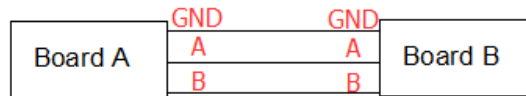
```

# cansend can0 123#DEADBEEF (Transmitter)

```

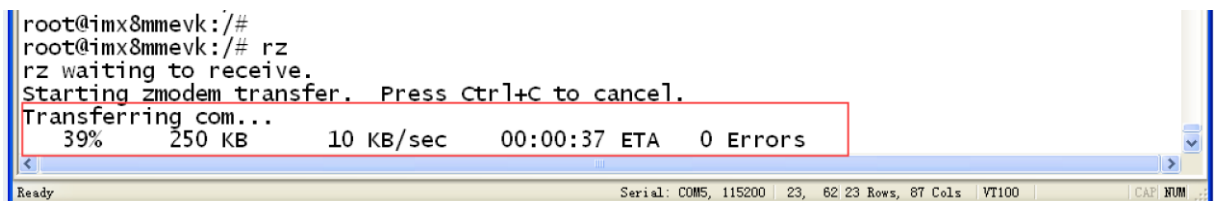
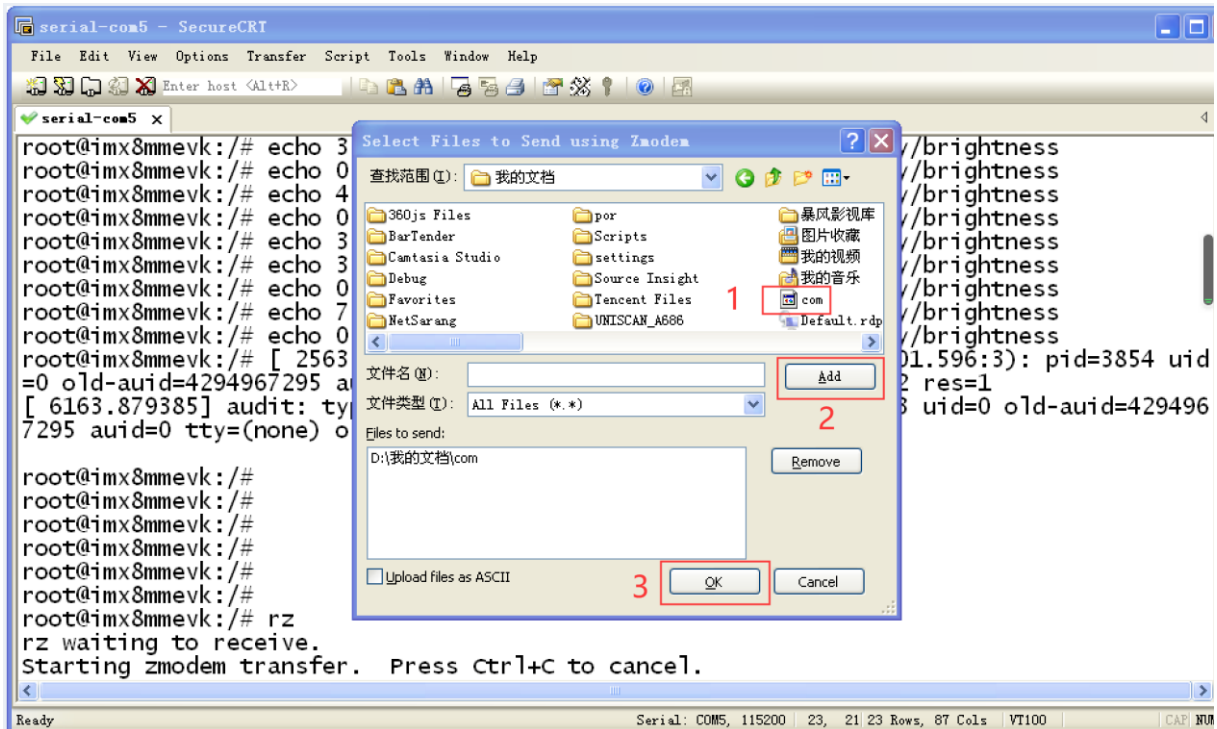
6.10 RS485

Connect the RS485 ports of Board A and B with the test line.



For Board A, execute the follow commands at **Serial terminal A** to set RS485 as Receiver.

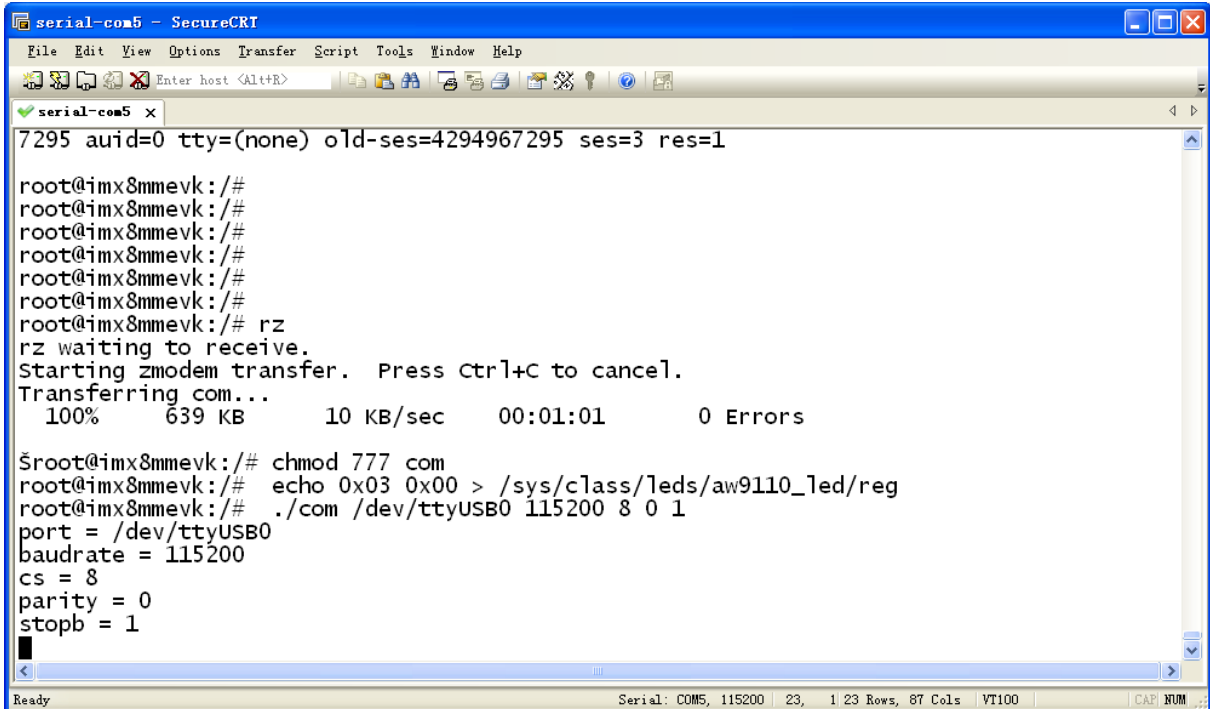
rz (send "com" file)



chmod 777 com

echo 0x03 0x00 > /sys/class/leds/aw9110_led/reg (Set as Receiver)

./com /dev/ttyUSB0 115200 8 0 1



```

serial-com5 - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
serial-com5 x
7295 auid=0 tty=(none) old-ses=4294967295 ses=3 res=1

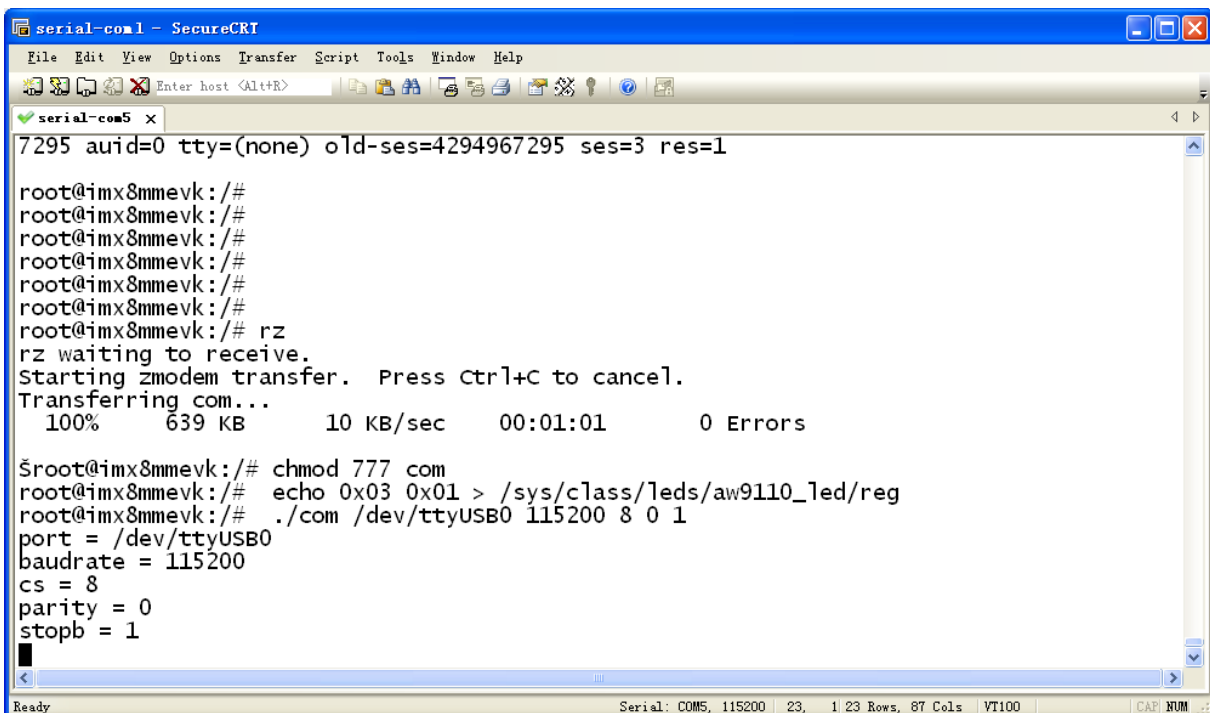
root@imx8mmevk:/#
root@imx8mmevk:/#
root@imx8mmevk:/#
root@imx8mmevk:/#
root@imx8mmevk:/#
root@imx8mmevk:/#
root@imx8mmevk:/# rz
rz waiting to receive.
Starting zmodem transfer. Press Ctrl+C to cancel.
Transferring com...
  100%   639 KB   10 KB/sec   00:01:01   0 Errors

$root@imx8mmevk:/# chmod 777 com
root@imx8mmevk:/# echo 0x03 0x00 > /sys/class/leds/aw9110_led/reg
root@imx8mmevk:/# ./com /dev/ttyUSB0 115200 8 0 1
port = /dev/ttyUSB0
baudrate = 115200
cs = 8
parity = 0
stopb = 1
█
Ready Serial: COM5, 115200 23, 1 23 Rows, 87 Cols VT100 CAP NUM
  
```

For Board B, execute the follow commands at **Serial terminal B** to set RS485 as Transmitter.

```

# rz (send "com" file)
#chmod 777 com
# echo 0x03 0x01 > /sys/class/leds/aw9110_led/reg (Set as Transmitter)
# ./com /dev/ttyUSB0 115200 8 0 1
  
```



```

serial-com1 - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
serial-com5 x
7295 auid=0 tty=(none) old-ses=4294967295 ses=3 res=1

root@imx8mmevk:/#
root@imx8mmevk:/#
root@imx8mmevk:/#
root@imx8mmevk:/#
root@imx8mmevk:/#
root@imx8mmevk:/#
root@imx8mmevk:/# rz
rz waiting to receive.
Starting zmodem transfer. Press Ctrl+C to cancel.
Transferring com...
  100%   639 KB   10 KB/sec   00:01:01   0 Errors

$root@imx8mmevk:/# chmod 777 com
root@imx8mmevk:/# echo 0x03 0x01 > /sys/class/leds/aw9110_led/reg
root@imx8mmevk:/# ./com /dev/ttyUSB0 115200 8 0 1
port = /dev/ttyUSB0
baudrate = 115200
cs = 8
parity = 0
stopb = 1
█
Ready Serial: COM5, 115200 23, 1 23 Rows, 87 Cols VT100 CAP NUM
  
```

The Transmitter and receiver can be converted by execute the command

```

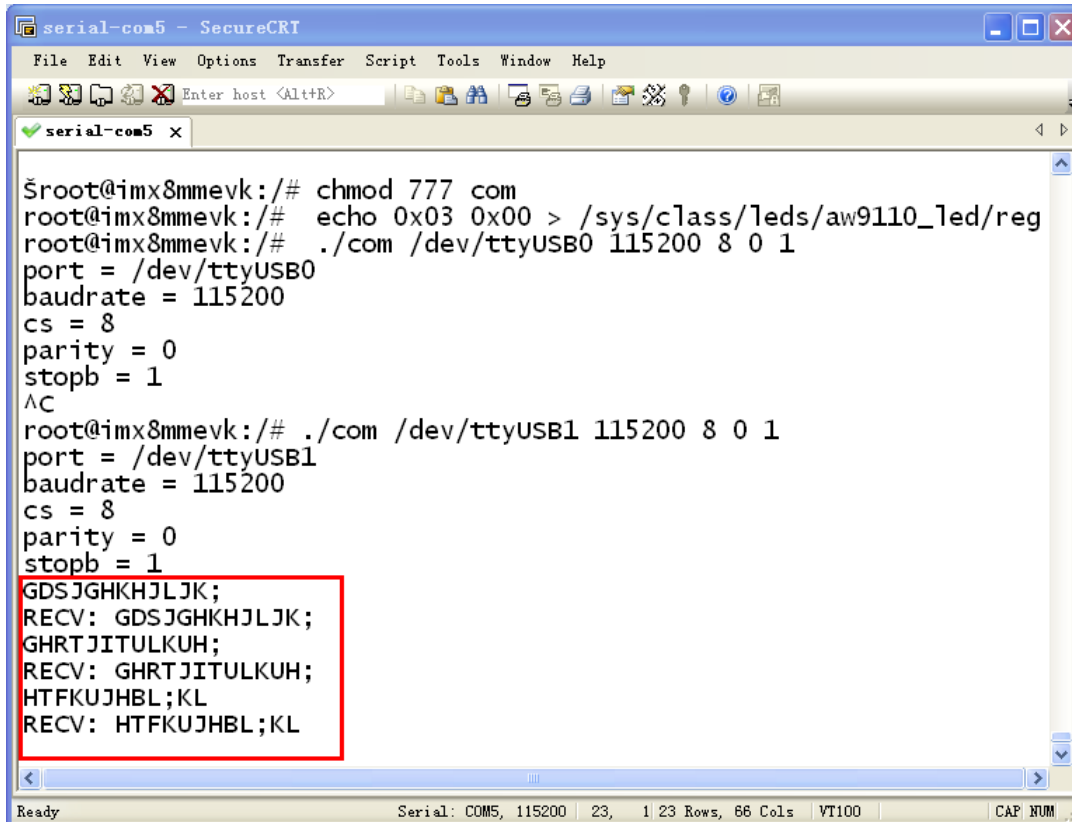
# echo 0x03 0x00 > /sys/class/leds/aw9110_led/reg (Set as Receiver)
# ./com /dev/ttyUSB0 115200 8 0 1
or
# echo 0x03 0x01 > /sys/class/leds/aw9110_led/reg (Set as Transmitter)
# ./com /dev/ttyUSB0 115200 8 0 1
  
```

6.11 UART(J14, COM1)

Connect RX&TX (PIN2&3 of J14/COM1), then execute the commands to run the test program at serial terminal.

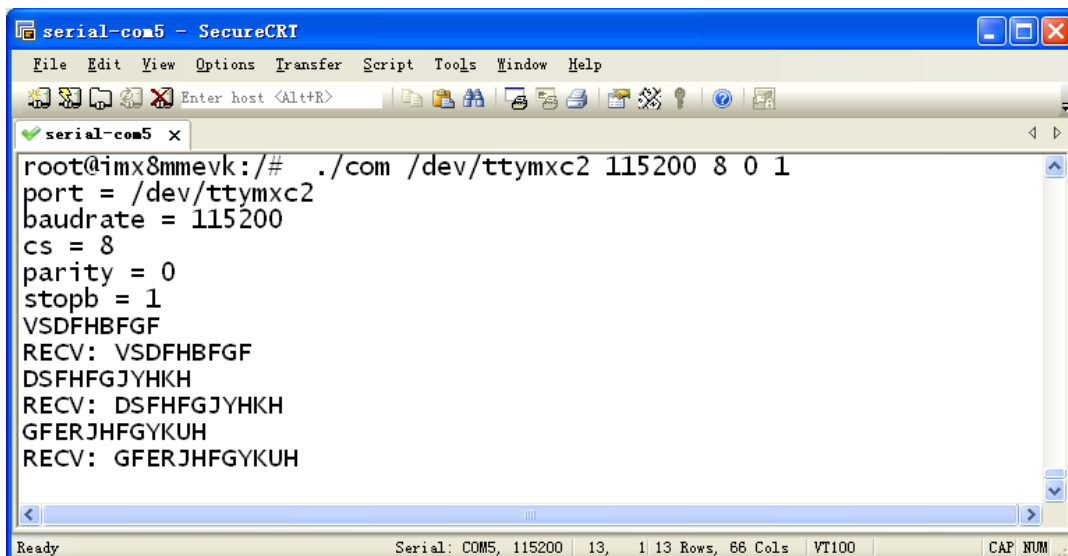
```
# ./com /dev/ttyUSB1 115200 8 0 1
```

```
# ./com /dev/ttymxc2 115200 8 0 1
```



```

serial-com5 - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
serial-com5 x
root@imx8mmevk:/# chmod 777 com
root@imx8mmevk:/# echo 0x03 0x00 > /sys/class/leds/aw9110_led/reg
root@imx8mmevk:/# ./com /dev/ttyUSB0 115200 8 0 1
port = /dev/ttyUSB0
baudrate = 115200
cs = 8
parity = 0
stopb = 1
^C
root@imx8mmevk:/# ./com /dev/ttyUSB1 115200 8 0 1
port = /dev/ttyUSB1
baudrate = 115200
cs = 8
parity = 0
stopb = 1
GDSJGHKHJLJK;
RECV: GDSJGHKHJLJK;
GHRTJITULKUH;
RECV: GHRTJITULKUH;
HTFKUJHBL;KL
RECV: HTFKUJHBL;KL
Ready Serial: COM5, 115200 | 23, 1 23 Rows, 66 Cols VT100 CAP NUM
  
```



```

serial-com5 - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
serial-com5 x
root@imx8mmevk:/# ./com /dev/ttymxc2 115200 8 0 1
port = /dev/ttymxc2
baudrate = 115200
cs = 8
parity = 0
stopb = 1
VSDFBFGF
RECV: VSDFBFGF
DSFHFGJYHKH
RECV: DSFHFGJYHKH
GFERJHFGYKUH
RECV: GFERJHFGYKUH
Ready Serial: COM5, 115200 | 13, 1 13 Rows, 66 Cols VT100 CAP NUM
  
```