

“Offline” Editor For Linux Sdcard Mirror

Biyong Sun

REV 2.0
15 SEP 2022



EXTERNAL USE



SECURE CONNECTIONS
FOR A SMARTER WORLD

Any support, information, and technology (“Materials”) provided by NXP are provided AS IS, without any warranty express or implied, and NXP disclaims all direct and indirect liability and damages in connection with the Material to the maximum extent permitted by the applicable law. NXP accepts no liability for any assistance with applications or product design. Materials may only be used in connection with NXP products. Any feedback provided to NXP regarding the Materials may be used by NXP without restriction.

Purpose

Background in Field

- Under some situations, need to modify the binary image.
- Under some situations, no development environment.
- Under some situations, even no Linux. Only have windows.
- Board bring up need simple settings, such as simple/tiny rootfs.

Key Features

Features

- Create small sdcard mirror image
- All operations are on binary files no need development environment(offline)
- Set the u-boot environment on binary image
- Windows OS support

Create Sdcard Mirror Ext

The Demo Environment

The default demo image files are from L5.4.70-2.3.0_images_MX8MMEVK.zip.
The hardware is i.MX8MM LPDDR4 board(PMIC bd71837).

Introduction create_sdcard_mirror_ext

```
create_sdcard_mirror_ext
| a.sdcard                --- sdcard mirror (output)
| create_sd_mirror_i.MX8MQ_i.MX8MM.bat    --- batch for creating MX8MQ /MX8MM sdcard mirror
| create_sd_mirror_i.MX8_i.MX8X_i.MX8MN_i.MX8MP_iMX8ULP.bat --- batch for creating MX8/MX8X/MX8MN/MX8MP/iMX8ULP sdcard mirror
| flash.bin              --- flash.bin (input)
| fw_env.config          --- configuration file for fw_printenv, fw_setenv
| Image                 --- linux kernel image
| imx8mm-evk.dtb        --- device tree with PCA9450AAHN pmic
| imx8mm-evk-bd71837.dtb --- device tree with bd71837 pmic
| modify_sd_mirror.bat  --- batch file for modifying sdcard mirror
| readme.txt            --- readme
| u-boot-env            --- u-boot environment file (input optional)
+---doc
|   Offline Editor for Linux sdcard mirror.pdf    --- this doc
|
+---partitons_mirrors      --- partition mirrors for creating sdcard mirror

\---tools
  +---gptfdisk_win
  +---misc
  +---mtools_win
  +---truncate_win
  +---u-boot_env_tools_win
  \---win32diskimager-binary
8   EXTERNAL USE
```



Create Sdcard Mirror

please provide:

flash.bin --- bootloader
Image --- kernel image
dtb --- device tree such as imx8mm-evk.dtb, imx8qxp-mek.dtb, etc.
fw_env.config --- configuration file for fw_printenv, fw_setenv (optional)
u-boot-env --- could be the initial environment file by "make u-boot-initial-env" (optional)

Double click one of the following batch file.
Will create a.sdcard as output sdcard mirror.

create_sd_mirror_i.MX8MQ_i.MX8MM.bat
create_sd_mirror_i.MX8_i.MX8X_i.MX8MN_i.MX8MP_iMX8ULP.bat

Note:

fw_env.config and u-boot-env are optional.
If **both** fw_env.config and u-boot-env exist, it will set u-boot environment.

Create Sdcard Mirror(cont.)

fw_env.config:

```
#a.sdcard 0x400000 0x4000 #8MQ,8MM, 8MN, 8MP since lf-5.10.52-2.1.0
```

```
#a.sdcard 0x400000 0x2000 #8, 8X,8ULP
```

```
a.sdcard 0x400000 0x1000 #8MQ,8MM, 8MN, 8MP
```

Note: Please check the u-boot environment offset from different BSP releases.

u-boot-env:

could be the initial environment file by "make u-boot-initial-env"

In the demo, using a tricky way to have the "u-boot-initial-env" runtime then set fdt_file to imx8mm-evk-bd71837.dtb and save.

```
bootcmd=echo first run...;env default -a;setenv fdt_file imx8mm-evk-bd71837.dtb;savee;reset
```

Create Sdcard Mirror(cont.)

Double click create_sd_mirror_i.MX8MQ _i.MX8MM.bat

After finishing running batch file, you will find file a.sdcard as output for sdcard mirror.

Modify Sdcard Mirror

Modify Sdcard Mirror

C:\WINDOWS\system32\cmd.exe

```
-----  
| 32k/33k | u-boot | boot(fat) | rootfs |  
-----  
~
```

fat offset of sdcard mirror(default 8M)

```
sd.hlp --- This help.  
sd.fat.ls <sdcard mirror> --- List files in fat.  
sd.fat.setoffset <offset> --- Set fat offset of sdcard mirror.  
sd.fat.offset --- Display fat offset settings.  
sd.fat.upld <sdcard mirror> <local file> [file in fat] --- Copy local file to boot fat partition.  
ex: sd.fat.upld a.sdcard Image  
sd.fat.upld a.sdcard Image Image1  
sd.fat.dnld <sdcard mirror> <file in fat> [local file] --- Copy file in boot fat partition to local.  
ex: sd.fat.dnld a.sdcard Image .  
sd.fat.dnld a.sdcard Image Image1  
sd.fat.del <sdcard mirror> <file in fat> --- Delete file in boot fat partition.  
ex: sd.fat.del a.sdcard Image  
sd.fw.pri --- Print u-boot environment by fw_printenv.  
sd.fw.set --- Set u-boot environment by fw_setenv.  
sd.fw.env.edit --- Modify u-boot environment by notepad.  
sd.fw.conf.edit --- Modify fw_printenv/fw_setenv by notepad.  
sd.part.pri <sdcard mirror> --- List scard mirror partition information.  
sd.boot.mirror.32k <bootloader> --- Make bootloarder(flash.bin) only sdcard mirror(offset 32k).  
i.MX8|i.MX8X|i.MX8MN|i.MX8MP|iMX8SULP  
sd.boot.mirror.33k <bootloader> --- Make bootloarder(flash.bin) only sdcard mirror(offset 33k).  
i.MX8MQ|i.MX8MM  
sd.gpt.recover <sdcard mirror> --- load main partition table from disk (rebuilding backup).  
sd.diskimager --- Call Win32DiskImager
```

Help



Help

1. double click modify_sd_mirror.bat
2. run sd.hlp

```
C:\WINDOWS\system32\cmd.exe
32k/33k | u-boot | boot(fat) | rootfs |
-----
fat offset of sdcard mirror(default 8M)

sd.hlp
sd.fat.ls      <sdcard mirror>
sd.fat.setoffset <offset>
sd.fat.offset
sd.fat.upld <sdcard mirror> <local file> [file in fat]
sd.fat.dnld <sdcard mirror> <file in fat> [local file]
sd.fat.del <sdcard mirror> <file in fat>
sd.fw.pri
sd.fw.set
sd.fw.env.edit
sd.fw.conf.edit
sd.part.pri <sdcard mirror>
sd.boot.mirror.32k <bootloader>
sd.boot.mirror.33k <bootloader>
sd.gpt.recover <sdcard mirror>
sd.diskimager

--- This help.
--- List files in fat.
--- Set fat offset of sdcard mirror.
--- Display fat offset settings.
--- Copy local file to boot fat partition.
ex: sd.fat.upld a.sdcard Image
sd.fat.upld a.sdcard Image Image1
--- Copy file in boot fat partition to local.
ex: sd.fat.dnld a.sdcard Image .
sd.fat.dnld a.sdcard Image Image1
--- Delete file in boot fat partition.
ex: sd.fat.del a.sdcard Image
--- Print u-boot environment by fw_printenv.
--- Set u-boot environment by fw_setenv.
--- Modify u-boot environment by notepad.
--- Modify fw_printenv/fw_setenv by notepad.
--- List scard mirror partition information.
--- Make bootloarder(flash.bin) only sdcard mirror(offset 32k).
i.MX8|i.MX8X|i.MX8MN|i.MX8MP|iMX8ULP
--- Make bootloarder(flash.bin) only sdcard mirror(offset 33k).
i.MX8MQ|i.MX8MM
--- load main partition table from disk (rebuilding backup).
--- Call Win32DiskImager
```


FAT On SDCARD

Fat On SDCARD

List file in fat partition:

1. double click modify_sd_mirror.bat
2. run sd.fat.ls a.sdcard

```
Volume in drive : has no label
Volume Serial Number is 7633-D85C
Directory for ::/

IMAGE          28013056 2022-06-09 17:08 Image
IMX8MM~1 DTB   45432 2022-06-09 17:08 imx8mm-evk-bd71837.dtb
IMX8MM~2 DTB   45667 2022-06-09 17:08 imx8mm-evk.dtb
  3 files          28 104 155 bytes
                   24 100 864 bytes free
```

Note: you can use `sd.part.pri <sdcard mirror>` to find the fat offset.
Then you can use `sd.fat.setoffset <offset>` to set the offset.
`sd.fat.offset` is a command to know the current fat offset.

Fat On SDCARD (Cont.)

Copy file from/to fat partition and delete file in fat partition:

Delete:

1. double click modify_sd_mirror.bat
2. run sd.fat.ls a.sdcard
3. run sd.fat.del a.sdcard imx8mm-evk-bd71837.dtb
4. run sd.fat.ls a.sdcard to check the result

Copy to fat partition:

1. double click modify_sd_mirror.bat
2. run sd.fat.ls a.sdcard
3. run sd.fat.upld a.sdcard imx8mm-evk-bd71837.dtb
4. run sd.fat.ls a.sdcard to check the result

Copy to fat partition:

1. double click modify_sd_mirror.bat
2. run sd.fat.ls a.sdcard
3. run sd.fat.dnld a.sdcard imx8mm-evk.dtb ■
4. run sd.fat.ls a.sdcard to check the result

U-BOOT Environment

U-boot Environment

double click modify_sd_mirror.bat

sd.fw.pri
sd.fw.set
sd.fw.env.edit
sd.fw.conf.edit

--- Print u-boot environment by fw_printenv.
--- Set u-boot environment by fw_setenv.
--- Modify u-boot environment by notepad.
--- Modify fw_printenv/fw_setenv by notepad.

Bootloader Only Sdcard Mirror

Bootloader Only Sdcard Mirror

```
sd.boot.mirror.32k <bootloader>    ---  i.MX8|i.MX8X|i.MX8MN|i.MX8MP|iMX8ULP
sd.boot.mirror.33k <bootloader>    ---  i.MX8MQ|i.MX8MM
```

sd.boot.mirror.32k imx-boot-imx8q MPC0mek-sd.bin-flash

You will find imx-boot-imx8q MPC0mek-sd.bin-flash.32k.sdcard mirror.

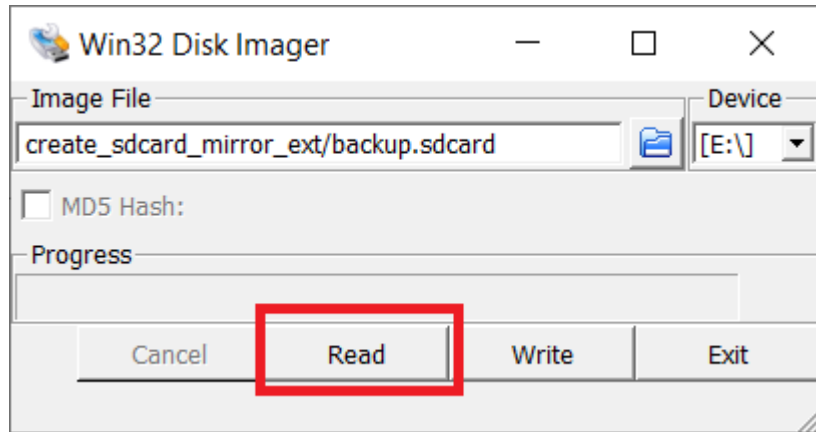
sd.boot.mirror.33k imx-boot-imx8m mevk-sd.bin-flash_evk

You will find imx-boot-imx8m mevk-sd.bin-flash_evk.33k.sdcard.

Backup SDCARD

Backup MBR SDCARD

1. double click modify_sd_mirror.bat
2. run sd.diskimager to call “Win32 Disk Imager” to dump sdcard(backup.sdcard)



Backup MBR SDCARD(Cont.)

3. run modify_sd_mirror.bat
4. sd.part.pri backup.sdcard to check the used size.

```
*****
Found invalid GPT and valid MBR; converting MBR to GPT format
in memory.
*****

Disk backup.sdcard: 31116287 sectors, 14.8 GiB
Sector size (logical): 512 bytes
Disk identifier (GUID): 7720AA65-332B-4231-9F3A-1F9E7936E48A
Partition table holds up to 128 entries
Main partition table begins at sector 2 and ends at sector 33
First usable sector is 34, last usable sector is 31116253
Partitions will be aligned on 2048-sector boundaries
Total free space is 25747806 sectors (12.3 GiB)

Number  Start (sector)    End (sector)  Size      Code  Name
  1         16384             186775    83.2 MiB   0700  Microsoft basic data
  2        196608           5394629    2.5 GiB   8300  Linux filesystem
```



Backup MBR SDCARD (Cont.)

5. use truncate to 2.6G

$5394629 / 2 = 2697314.5$ K. Use 2697315K to hold.

`truncate -s 2697315K backup.sdcard`

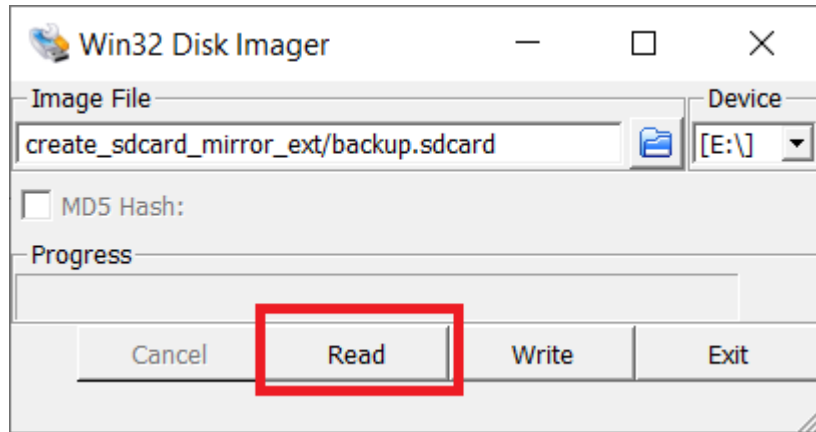
6. `sd.part.pri backup.sdcard` to double confirm the result.

```
Warning! Secondary partition table overlaps the last partition by
33 blocks!
You will need to delete this partition or resize it in another utility.
Disk backup.sdcard: 5394630 sectors, 2.6 GiB
Sector size (logical): 512 bytes
Disk identifier (GUID): 9B0935D4-E14F-457B-BC40-33C5249398C2
Partition table holds up to 128 entries
Main partition table begins at sector 2 and ends at sector 33
First usable sector is 34, last usable sector is 5394596
Partitions will be aligned on 2048-sector boundaries
Total free space is 26182 sectors (12.8 MiB)

Number  Start (sector)    End (sector)  Size      Code  Name
   1            16384             186775      83.2 MiB   0700  Microsoft basic data
   2           196608           5394629     2.5 GiB   8300  Linux filesystem
```

Backup GPT SDCARD

1. double click modify_sd_mirror.bat
2. run sd.diskimager to call “Win32 Disk Imager” to dump sdard(backup.sdcard)



Backup GPT SDCARD(Cont.)

3. run `modify_sd_mirror.bat`
4. `sd.part.pri backup.sdcard` to check the used size.

```
. Partition table scan:
  MBR: protective
  BSD: not present
  APM: not present
  GPT: present
-----
Found valid GPT with protective MBR; using GPT
Disk backup.sdcard: 31116287 sectors, 14.8 GiB
Sector size (logical): 512 bytes
Disk identifier (GUID): 5E8C4525-B960-4FB3-A29F-BC573E51B6E6
Partition table holds up to 128 entries
Main partition table begins at sector 2 and ends at sector 33
First usable sector is 34, last usable sector is 31116226
Partitions will be aligned on 2048-sector boundaries
Total free space is 26790817 sectors (12.8 GiB)

Number  Start (sector)    End (sector)  Size      Code  Name
-----  -
   1         16384             147455      64.0 MiB   0700  Microsoft basic data
   2        147456            4341759     2.0 GiB   8300  Linux filesystem
-----
```

Backup GPT SDCARD (Cont.)

5. use truncate to 2.1G

$4341759 / 2 = 2170879.5$ K. Use 2170880K to hold. And add 17K for second GPT at tail.

$2170880 + 17 = 2170897$ K

```
truncate -s 2170897K backup.sdcard
```

6. `sd.part.pri backup.sdcard`
to double confirm the result.

```
Partition table scan:
  MBR: protective
  BSD: not present
  APM: not present
  GPT: damaged
*****
Caution: Found protective or hybrid MBR and corrupt GPT. Using GPT, but disk
verification and recovery are STRONGLY recommended.
*****
Disk backup.sdcard: 4341794 sectors, 2.1 GiB
Sector size (logical): 512 bytes
Disk identifier (GUID): 5E8C4525-B960-4FB3-A29F-BC573E51B6E6
Partition table holds up to 128 entries
Main partition table begins at sector 2 and ends at sector 33
First usable sector is 34, last usable sector is 31116226
Partitions will be aligned on 2048-sector boundaries
Total free space is 26790817 sectors (12.8 GiB)

Number  Start (sector)    End (sector)  Size      Code  Name
   1        16384             147455     64.0 MiB   0700  Microsoft basic data
   2       147456             4341759    2.0 GiB   8300  Linux filesystem
```



Backup GPT SDCARD (Cont.)

7. Recover GPT

```
sd.gpt.recover backup.sdcard
```

```
Partition table scan:
  MBR: protective
  BSD: not present
  APM: not present
  GPT: present
  _____
Found valid GPT with protective MBR; using GPT.
Disk backup.sdcard: 4341794 sectors, 2.1 GiB
Sector size (logical): 512 bytes
Disk identifier (GUID): 5E8C4525-B960-4FB3-A29F-BC573E51B6E6
Partition table holds up to 128 entries
Main partition table begins at sector 2 and ends at sector 33
First usable sector is 34, last usable sector is 4341760
Partitions will be aligned on 2048-sector boundaries
Total free space is 16351 sectors (8.0 MiB)

Number  Start (sector)    End (sector)  Size      Code  Name
-----  -
1         16384             147455       64.0 MiB  0700  Microsoft basic data
2        147456            4341759      2.0 GiB   8300  Linux filesystem
```

Modify Device Tree

Check Device Tree

C:\WINDOWS\system32\cmd.exe

```
D:\create_sdcard_mirror_ext>fdtdump imx8mm-evk.dtb
```

```
**** fdtdump is a low-level debugging tool, not meant for general use.
**** If you want to decompile a dtb, you probably want
**** dtc -I dtb -O dts <filename>
```

```
/dts-v1/;
// magic:          0xd00dfeed
// totalsize:     0xb263 (45667)
// off_dt_struct: 0x38
// off_dt_strings: 0xa590
// off_mem_rsvmap: 0x28
// version:       17
// last_comp_version: 16
// boot_cpuid_phys: 0x0
// size_dt_strings: 0xcd3
// size_dt_struct: 0xa558

/ {
    compatible = "fsl,imx8mm-evk", "fsl,imx8mm";
    interrupt-parent = <0x00000001>;
    #address-cells = <0x00000002>;
    #size-cells = <0x00000002>;
    model = "FSL i.MX8MM EVK board";
    aliases {
        ethernet0 = "/soc@0/bus@30800000/ethernet@30be0000";
        i2c0 = "/soc@0/bus@30800000/i2c@30a20000";
        i2c1 = "/soc@0/bus@30800000/i2c@30a30000";
        i2c2 = "/soc@0/bus@30800000/i2c@30a40000";
        i2c3 = "/soc@0/bus@30800000/i2c@30a50000";
        serial0 = "/soc@0/bus@30800000/serial@30860000";
        serial1 = "/soc@0/bus@30800000/serial@30890000";
        serial2 = "/soc@0/bus@30800000/serial@30880000";
        serial3 = "/soc@0/bus@30800000/serial@30a60000";
        spi0 = "/soc@0/bus@30800000/spi@30820000";
        spi1 = "/soc@0/bus@30800000/spi@30830000";
        spi2 = "/soc@0/bus@30800000/spi@30840000";
        mmc0 = "/soc@0/bus@30800000/mmc@30b40000";
        mmc1 = "/soc@0/bus@30800000/mmc@30b50000";
```

C:\WINDOWS\system32\cmd.exe

```
D:\create_sdcard_mirror_ext>fdtget -p imx8mm-evk.dtb mmc2
```

```
compatible
reg
interrupts
clocks
clock-names
assigned-clocks
assigned-clock-rates
fsl,tuning-start-tap
fsl,tuning-step
bus-width
status
pinctrl-names
pinctrl-0
pinctrl-1
pinctrl-2
non-removable
```

```
D:\create_sdcard_mirror_ext>fdtget imx8mm-evk.dtb mmc2 status
okay
```

```
D:\create_sdcard_mirror_ext>fdtget imx8mm-evk.dtb mmc2 bus-width
8
```

```
D:\create_sdcard_mirror_ext>_
```



Modify Device Tree

C:\WINDOWS\system32\cmd.exe

```
D:\create_sdcard_mirror_ext>fdtget      imx8mm-evk.dtb  mmc2  status
okay

D:\create_sdcard_mirror_ext>fdtput -t s  imx8mm-evk.dtb  mmc2  status disabled

D:\create_sdcard_mirror_ext>fdtget      imx8mm-evk.dtb  mmc2  status
disabled

D:\create_sdcard_mirror_ext>
```



SECURE CONNECTIONS
FOR A SMARTER WORLD