

背景:

env 分区在读写过程中,如果突然断电,有极小的概率会出现数据损坏,导致 env 丢失,一旦数据丢失系统会启动失败,为保证系统的启动,当 env 分区数据 CRC 校验失败时,会自动从 default 值去启动,由于每个客户分区的设置可能不尽相同,只需要按照以下两步,在相应地方配置好默认值即可。

以下 log 代表 CRC 校验失败,会 load default 值

```
*** Warning - bad CRC, using default environment
```

这里拿 NAND flash 举例 (*NOR flash 的直接跳到第二步即可)

一、uboot 设置 default mtdpart: (分区信息)

需修改文件 boot/include/configs/infinity2m.h 中对应的宏定义 MTDPARTS_DEFAULT

*注意这里的 mtdpart 跟 uboot 下输入 mtdpart 后的保持一致:

Eg: uboot 下输入 mtdpart

```
SigmaStar # mtdpart
device nand0 <nand0>, # parts = 12
# name      size      offset      mask_flags
0: IPL0      0x00060000 0x00140000 0
1: IPL1      0x00060000 0x001a0000 0
2: IPL_CUST0 0x00060000 0x00200000 0
3: IPL_CUST1 0x00060000 0x00260000 0
4: UBOOT0    0x000c0000 0x002c0000 0
5: UBOOT1    0x000c0000 0x00380000 0
6: ENV       0x00060000 0x00440000 0
7: KEY_CUST  0x00020000 0x00460000 0
8: LOGO      0x00060000 0x004c0000 0
9: KERNEL    0x00500000 0x00520000 0
10: RECOVERY 0x00800000 0x00820000 0
11: UBI       0x070e0000 0x00f20000 0

active partition: nand0,0 - (IPL0) 0x00060000 @ 0x00140000

defaults:
mtdids = nand0=nand0
mtdparts = mtdparts=nand0:384k@1280k(IPL0),384k(IPL1),384k(IPL_CUST0),384k(IPL_CUST1),768k(UBOOT0),768k(UBOOT1),384k(ENV),0x20000(KEY_CUST),0x60000(LOGO),0x500000(KERNEL),0x500000(RECOVERY),-(UBI)
SigmaStar #
```

则在 boot/include/configs/infinity2m.h 中将 MTDPARTS_DEFAULT 设置为相同的值

(*注意该文件中有多个 MTDPARTS_DEFAULT 宏变量,不想区分的话可以所有 MTDPARTS_DEFAULT 都改一下)

```
0 TRIAL - [infinity2m.h (Y:\alkaid\boot\include\configs)]
Tools View Window Help
bin2rcf.cpp (Y:\IPL\IPL\make\bin2rcf) cdjpege.h (drivers\mstar\jpd\infinity2m\inc) cli_hush.c (common) cmd_mtdparts.c (common) environment.h (include) env_default.h (include) infinity2
199: #define CONFIG_CMD_SPINAND_CIS
200: #define CONFIG_CMD_UBI
201: /* #define CONFIG_CMD_UBIFS */
202: #define CONFIG_UBI_MWRITE
203: #define MTDIDS_DEFAULT "nand0=nand0" /* "nor0=physmap-flash.0,nand0=nand" */
204: /* must be different from real partition to test NAND partition function */
205: #define MTDPARTS_DEFAULT "mtdparts=nand0:384k@1280k(IPL0),384k(IPL1),384k(IPL_CUST0),384k(IPL_CUST1),768k(IPL_CUST0),768k(UBOOT0),768k(UBOOT1),384k(ENV),0x20000(KEY_CUST),0x60000(LOGO),0x500000(KERNEL),0x500000(RECOVERY),-(UBI)"
206: /* #define MTDPARTS_DEFAULT "mtdparts=nand0:0x60000@0x140000(IPL0),0x60000(IPL1),0x60000(IPL_CUST0),0x60000(IPL_CUST1),0x60000(IPL_CUST2),0x60000(IPL_CUST3),0x60000(IPL_CUST4),0x60000(IPL_CUST5),0x60000(IPL_CUST6),0x60000(IPL_CUST7),0x60000(IPL_CUST8),0x60000(IPL_CUST9),0x60000(IPL_CUST10),0x60000(IPL_CUST11),0x60000(IPL_CUST12)" */
207:
208:
209: #define CONFIG_EXTRA_ENV_SETTINGS
210: "mtdids=" MTDIDS_DEFAULT "\0"
211: "mtdparts=" MTDPARTS_DEFAULT "\0"
```

二、uboot 设置 Default Bootargs 跟 bootcommand:

目前新的版本中,在 Uboot/include/configs/infinity2m.h 文件末尾有两个宏定义:

Eg: uboot 输入 printenv bootargs/printenv bootcmd

```
SigmaStar # printenv bootcmd
bootcmd=bootlogo 0 0 0 0; mw 1f001cc0 11; gpio out 8 0; nand read.e 0x22000000 KERNEL 0x500000; gpio
SigmaStar # printenv bootargs
bootargs=console=ttyS0,115200 ubi.mtd=UBI,2048 root=ubi:rootfs ro rootfstype=ubifs init=/linuxrc rootwa
,max_start_off=0x3300000,max_end_off=0x3600000 mtdparts=nand0:384k@1280k(IPL0),384k(IPL1),384k(IPL_CUST0),384k(IPL_CUST1),768k(UBOOT0),768k(UBOOT1),384k(ENV),0x20000(KEY_CUST),0x60000(LOGO),0x500000(KERNEL),0x500000(RECOVERY),-(UBI)
SigmaStar #
```

将截图中两个 bootargs 跟 bootcmd 的值填到以下截图对应的 CONFIG_BOOTARGS/CONFIG_BOOTCOMMAND 即可

```
AL - [infinity2m.h (Y:\alkaid\boot\include\configs) *]
ols View Window Help
environment.h (include) env_common.c (common) env_flash.c (common) env_nand.c (common) env_sf.c (common) fw_
400: /* SENSOR */
401: #define CONFIG_MS_SRCFG
402:
403: /* default env */
404:
405: #define CONFIG_BOOTARGS "ttyS0,115200 ubi.mtd=UBI,2048 root=ubi:rootfs ro rd
406:
407: #define CONFIG_BOOTCOMMAND
408: "bootlogo 0 0 0 0 0; "
409: "mw 1f001cc0 11; " \
410: "gpio out 8 0; "
411: "nand read.e 0x22000000 KERNEL 0x500000; "
412: "gpio out 8 1; "
413: "bootm 0x22000000; "
414: "nand read.e 0x22000000 RECOVERY 0x500000; "
415: "bootm 0x22000000;"
416:
```

三、Uboot 设置自定义 Default ENV:

修改文件: boot\include\env_default.h

在 default_environment[] 数组中参照其它 ENV 添加自己的默认 ENV (*以上第一/第二步设置的 bootargs 跟 bootcmd 也是最终定义到这个数组的)

eg:

```
19: {
20: #elif defined(DEFAULT_ENV_INSTANCE_STATIC)
21: static char default_environment[] = {
22: #else
23: const uchar default_environment[] = {
24: #endif
25: "my_env=" "my_env_test" "\0"
26:
27: #ifdef CONFIG_ENV_CALLBACK_LIST_DEFAULT
28: ENV_CALLBACK_VAR "=" CONFIG_ENV_CALLBACK_LIST_DEFAULT
29: #endif
30: #ifdef CONFIG_ENV_FLAGS_LIST_DEFAULT
31: ENV_FLAGS_VAR "=" CONFIG_ENV_FLAGS_LIST_DEFAULT "\0"
```

```
SigmaStar # printenv my_env
my_env=my_env_test
```

四、验证方法:

只需要在 uboot 控制台下将 env 分区擦除, 重启即会 load default env

擦除 ENV 分区命令: nand erase.part ENV

```
SigmaStar # nand erase.part ENV  
  
NAND erase.part: device 0 offset 0x440000, size 0x60000  
Erasing at 0x480000 -- 100% complete.  
Time:20630 us, speed:19060 KB/s  
OK  
SigmaStar #
```

以上，在 boot/include/configs/infinity2m.h 设置了系统启动必须的 default env 之后，重编 uboot 并更新到板端，至少可以保证，env 数据丢失的时候，依旧可以正常启动到系统