

# 目次

ファームウェアの構造

1

**ストレージ(Flash Memory)の構成**

1

Nand Flashメモリの構成

1

SPI Flashメモリの構成

1

**ファイルシステムへのマウント**

2



# ファームウェアの構造

## ストレージ(Flash Memory)の構成

### Nand Flashメモリの構成

MA-E3xxシリーズには256MiBのNAND Flashが搭載されています。  
構成は、下表のようになっています。

デバイス	サイズ	名前	備考
mtdd0	255MiB	UBI	UBI <sup>1)</sup>
mtdd1	1MiB	OOPS	OOPSメッセージ保存用
以下UBI領域内のボリューム			
mtdd2	4158KiB	kernel.0	カーネル [0]
mtdd3	4158KiB	kernel.1	カーネル [1]
mtdd4	2142KiB	initramfs.0	initramfs [0]
mtdd5	2142KiB	initramfs.1	initramfs [1]
mtdd6	2142KiB	kmod.0	カーネルモジュール [0]
mtdd7	2142KiB	kmod.1	カーネルモジュール [1]
mtdd8	82026KiB	rootfs.0	root filesystem [0]
mtdd9	82026KiB	rootfs.1	root filesystem [1]
mtdd10	25704KiB	opt	Java SE Embedded 8
mtdd11	44856KiB	overlay	変更分保存用領域(オーバーレイ領域)

### SPI Flashメモリの構成

ブートデバイス用として2MiBのSPI Flashメモリが搭載されています。  
構成は、下表のようになっています。

デバイス	サイズ	名前	備考
mtdd12	128KiB	MLO	bootloader 1段目
mtdd13	512KiB	barebox.bin	bootloader 本体
mtdd14	128KiB	barebox.env	bootloader 保存領域
mtdd15	128KiB	dtb.0	DeviceTree blob [0]
mtdd16	128KiB	dtb.1	DeviceTree blob [1]
mtdd17	128KiB	hist_firmware	ファームウェア更新履歴
mtdd18	896KiB	hist_boot	起動履歴

カーネルinitramfsカーネルモジュールroot filesystemDeviceTree blob に “ [0] ” / “ [1] ” をつけているとおり、それぞれを2セット保持し、ファームウェア更新などで万が一起動しなくなった場合にもう片方のセットで起動することができるようになっています。

# ファイルシステムへのマウント

ディレクトリ構造は下記のようにになっています。

```
/ (aufs) --+
|
+- ro/ (squashfs) +- lib/modules (squashfs)
|                  |
|                  +- usr/lib/jvm (squashfs)
|
+- rw/ (tmpfs/ext4)
|
+- rw.tmpfs/ (tmpfs)
```

initramfs の [scripts/init-bottom/\\_rootoverlay](#) スクリプトにより、下記シーケンスでmount処理が行われます。

1. (initramfs initスクリプトによりUBIの “rootfs.x” 領域が  $\${rootmnt}$  へ mount される。)
2. /overlay, /ro, /rw, /rw.tmpfs のディレクトリを作成 (mkdir)
3. tmpfs を /rw.tmpfs へ mount
4. tmpfs を /rw へ mount
5. UBI の “kmod.x” 領域を /rw.tmpfs へコピーし、 $\${rootmnt}$ /lib/modules へ mount
6. UBI の “opt” 領域を /rw.tmpfs へコピーし、 $\${rootmnt}$ /usr/lib/jvm へ mount
7.  $\${rootmnt}$  を /ro へ 移動 (mount -move オプションにより)
8. /ro と /rw を **aufs**<sup>2)3)</sup> でまとめて /overlay へ mount
9. /rw.tmpfs に、var/log ディレクトリを作成 (syslog領域用)
10. /rw.tmpfs/var/log と /ro/var/log を **aufs** でまとめて /overlay/var/log へ mount
11. /rw/lib/modules と /ro/lib/modules を **aufs** でまとめて /overlay/lib/modules へ mount
12. /rw/usr/lib/jvm と /ro/usr/lib/jvm を **aufs** でまとめて /overlay/usr/lib/jvm へ mount
13. /ro を /overlay/ro へ 移動 (mount -move オプションにより)
14. /rw を /overlay/rw へ 移動 (mount -move オプションにより)
15. /rw.tmpfs を /overlay/rw.tmpfs へ 移動 (mount -move オプションにより)
16. /overlay を  $\${rootmnt}$  へ 移動 (mount -move オプションにより)

その後、init スクリプトにより  $\${rootmnt}$  ⇒ / へと rootfs の switch が行われ/sbin/init が起動されます。

1)

[UBI - Unsorted Block Images](#)

2)

<http://aufs.sourceforge.net/>

3)

<http://ja.wikipedia.org/wiki/Aufs>

From:

<https://ma-tech.centurysys.jp/> - **MA-X/MA-S/MA-E/IP-K Developers' WiKi**

Permanent link:

[https://ma-tech.centurysys.jp/doku.php?id=mae3xx\\_ope:firmware\\_structure:start&rev=1545954156](https://ma-tech.centurysys.jp/doku.php?id=mae3xx_ope:firmware_structure:start&rev=1545954156)

Last update: **2018/12/28 08:42**