

VIOカーネルドライバ仕様案1: V4L2

V4L2 MEM2MEMドライバ + Media Controller

- VIO制御をV4L2ドライバ(/dev/videoXで入出力)として実装
- RPF, WPF等各ユニットはV4L2 subdev (/dev/v4l-subdevXで設定)
- Media Controllerを使って(/dev/mediaXを介して)、各ユニット間のリンクを設定



mainline化が可能



他のカーネルデバイス(カメラ, LCD等)との連携が容易



複数タスク対応(排他制御)が容易



V4L2 MEM2MEMは2.6.35、Media Controllerは2.6.39でマージ

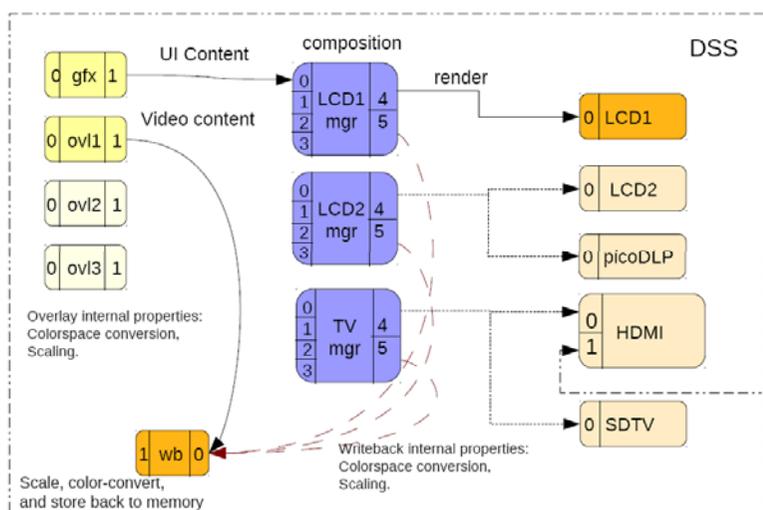


システムコール・オーバヘッドの影響を要検討

参考: V4L2 Media Controller

メディアデバイス内(間)のトポロジをユーザランドから設定可能

MC - DSS: Writeback (mem-2-mem)



MC - DSS: Writeback (mem-2-mem)

```

struct media_link_desc link;

link.source.entity = OMAP_DSS_ENTITY_OVERLAY0;
link.source.index = 1;
link.sink.entity = OMAP_DSS_ENTITY_MGR_LCD1;
link.sink.index = 0;
link.flags = MEDIA_LNK_FL_ENABLED;
ioctl(fd, MEDIA_IOC_SETUP_LINK, &link);

link.source.entity = OMAP_DSS_ENTITY_MGR_LCD1;
link.source.index = 4;
link.sink.entity = OMAP_DSS_ENTITY_PANEL_LCD1;
link.sink.index = 0;
link.flags = MEDIA_LNK_FL_ENABLED;
ioctl(fd, MEDIA_IOC_SETUP_LINK, &link);

/* Writeback path */
link.source.entity = OMAP_DSS_ENTITY_OVERLAY1;
link.source.index = 1;
link.sink.entity = OMAP_DSS_ENTITY_WRITEBACK;
link.sink.index = 0;
link.flags = MEDIA_LNK_FL_ENABLED;
ioctl(fd, MEDIA_IOC_SETUP_LINK, &link);

```

Sumit Semwal, TI, "Media Controller (MC) and OMAP2+ Display Subsystem", ELC2011より引用

VIOカーネルドライバ仕様案2: UIO

UIOドライバ

- VIOレジスタ領域をUIO経由でユーザランドへ
- ミドルウェア(ユーザランド・ライブラリ)でRPF, WPF等各ユニットのアプリケーション・レイヤを実装し、VIOを制御

-  レジスタ領域の定義のみmainline化
-  システムコール・オーバヘッドが最小限
-  V4L2 Media Controllerのようなリンク設定のためレイヤを実装要
-  各ユニット別の排他制御をユーザランドで実現する必要あり
-  他のカーネルドライバとの連携は個別対応(例:カメラは、V4L2 videobuf USERPTR経由)