Freescale Semiconductor

Document Number: IMX6FBAPI
Rev. L3.0.35_4.1.0, 09/2013

i.MX 6 Framebuffer API

Contents

List of Tables

1 Overview

The graphics software includes i.MX 6 Framebuffer (FB) API which enables users to easily create and port their graphics applications by using a framebuffer device without the need to expend additional effort handling platform-related tasks. i.MX 6 Framebuffer API focuses on providing mechanisms for controlling display, window, and pixmap render surfaces.

The EGL Native Platform Graphics Interface provides mechanisms for creating rendering surfaces onto which client APIs can draw, creating graphics contexts for client APIs, and synchronizing drawing by client APIs as well as native platform rendering APIs. This enables seamless rendering using Khronos APIs such as OpenGL ES and OpenVG for high-performance, accelerated, mixed-mode 2D, and 3D rendering. For further information on EGL, see http://www.khronos.org/registry/egl. The API described in this document is compatible with EGL version 1.4 of the specification.

This API document is current as of the software version specified in the document revision history. The following platforms are supported:

- Linux /X11
- Android
- Windows Embedded Compact

2 API Data Types and Environment Variables

2.1 Data Types

The GPU software provides platform independent member definitions for the following EGL types:

typedef struct _FBDisplay * EGLNativeDisplayType;
typedef struct _FBWindow * EGLNativeWindowType;
typedef struct _FBPixmap * EGLNativePixmapType;

Table 1. Types as listed on EGL 1.4 API Quick Reference Card
(from http://www.khronos.org/files/egl-1-4-quick-reference-card.pdf)

Types [2.1.1]

unsigned int	EGLBoolean
unsigned int	EGLenum
void	*EGLConfig
void	*EGLContext
void	*EGLDisplay
void	*EGLSurface
void	*EGLClientBuffer

The following types differ based on platform.

Windows platform:

HDC	EGLNativeDisplayType
HBITMAP	EGLNativePixmapType
HWND	EGLNativeWindowType

Linux/X11 platform:

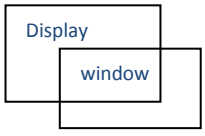
Display	*EGLNativeDisplayType
Pixmap	EGLNativePixmapType
Window	EGLNativeWindowType

Android platform:

ANativeWindow*	EGLNativeWindowType

2.2 Environment Variables

Table 2. i.MX 6 FB API Environment Variables

Environment Variables	Description
FB_MULTI_BUFFER	To use multiple-buffer rendering, set the environment variable **FB_MULTI_BUFFER** to an unsigned integer value, which indicates the number of buffers required. Maximum is 3. **Recommended values: 2 or 3.** The FB_MULTI_BUFFER variable can be set to any positive integer value. • If set to 1, the multiple buffer function is not enabled. • If set to 2 or 3, the driver will run as users expect. • If set to a value more than 3, the driver will use 3 as the buffer count.
FB_FRAMEBUFFER_0, FB_FRAMEBUFFER_1, FB_FRAMEBUFFER_2, FB_FRAMEBUFFER_n	To open a specified framebuffer device, set the environment variable **FB_FRAMEBUFFER_n** to a proper value (for example, FB_FRAMEBUFFER_0 = /dev/fb0). **Allowed values for n: any positive integer.** Note: If there are no environment variables set, the driver will try to use the default framebuffer devices (fb0 for index 0, fb1 for index 1, fb2 for index 2, fb3 for index 3, and so on).
FB_IGNORE_DISPLAY_SIZE	When set to a positive integer and a window's initial size request is greater than the display size, the window size will not be reduced to fit within the display. Global. **Allowed values: any positive integer.** Note: The drivers will read the value from this environment variable as a Boolean to check if the user wants to ignore the display size when creating a window. • If the variable is set to value, 0, or this environment variable is not set, then the driver when creating window will use display size to cut down the size of the window to ensure that the entire window area is inside the display screen. • If the user sets this variable to 1, or any positive integer value, then the window area can be partly or entirely outside of the display screen area (see the image below in which the ignore display size is equal to 1). 

Below are some usage syntax examples for environment variables:

To create a window with its size different from the display size, use the environment variable **FB_IGNORE_DISPLAY_SIZE**. Example usage syntax:

```
export FB_IGNORE_DISPLAY_SIZE=1
```

To let the driver use multiple buffers to do swap work, use the environment variable **FB_MULTI_BUFFER.** Example usage syntax:

```
export FB_MULTI_BUFFER=2
```

To specify the display device, use the environment variable **FB_FRAMEBUFFER_n**, where n = any positive integer. Example usage syntax:

```
export FB_FRAMEBUFFER_0=/dev/fb0
export FB_FRAMEBUFFER_1=/dev/fb1
export FB_FRAMEBUFFER_2=/dev/fb2
export FB_FRAMEBUFFER_3=/dev/fb3
```

3 API Description and Syntax

fbGetDisplay

Description:
This function is used to get the default display of the framebuffer device.

To open the framebuffer device, set an environment variable **FB_FRAMEBUFFER_n** to the framebuffer location.

Syntax:

```
EGLNativeDisplayType
fbGetDisplay (
      void *        context
);
```

Parameters:

context	Pointer to the native display instance.

Return Values:
The function returns a pointer to the EGL native display instance if successful; otherwise, it returns a NULL pointer.

fbGetDisplayByIndex

Description:
This function is used to get a specified display within a multiple framebuffer environment by providing an index number.

To use multiple buffers when rendering, set the environment variable **FB_MULTI_BUFFER** to an unsigned integer value, which indicates the number of buffers. Maximum is 3.

To open a specific framebuffer device, set environment variables to their proper values (e.g., set FB_FRAMEBUFFER_0 = /dev/fb0). If there are no environment variables set, the driver will try to use the default fb devices (fb0 for index 0, fb1 for index 1, fb2 for index 2, fb3 for index 3, and so on).

Syntax:

```
EGLNativeDisplayType
fbGetDisplayByIndex (
        int             DisplayIndex
);
```

Parameters:

DisplayIndex — An integer value where the integer is associated respectively with one of the following environment variables for framebuffer devices:
FB_FRAMEBUFFER_0
FB_FRAMEBUFFER_1
FB_FRAMEBUFFER_2
FB_FRAMEBUFFER_n

Return Value:
The function returns a pointer to the EGL native display instance if successful; otherwise, it returns a NULL pointer.

fbGetDisplayGeometry

Description:
This function is used to get display width and height information.

Syntax:

```
void
fbGetDisplayGeometry (
        EGLNativeDisplayType    Display,
        int *                   Width,
        int *                   Height
);
```

Parameters:

Display	[in] Pointer to EGL native display instance created by **fbGetDisplay.**
Width	[out] Pointer that receives the width of the display.
Height	[out] Pointer that receives the height of the display.

fbGetDisplayInfo

Description:
This function is used to get display information.

Syntax:

```
void
fbGetDisplayInfo (
        EGLNativeDisplayType    Display,
        int *                   Width,
        int *                   Height,
        unsigned long *         Physical,
        int *                   Stride,
        int *                   BitsPerPixel
);
```

Parameters:

Display	[in] A pointer to the EGL native display instance created by **fbGetDisplay.**
Width	[out] A pointer to the location that contains the width of the display.
Height	[out] A pointer to the location that contains the height of the display.
Physical	[out] A pointer to the location that contains the physical start address of the display.
Stride	[out] A pointer to the location that contains the stride of the display.
BitsPerPixel	[out] A pointer to the location that contains the pixel depth of the display.

fbDestroyDisplay

Description:
This function is used to destroy a display.

Syntax:

```
void
fbDestroyDisplay (
      EGLNativeDisplayType      Display
);
```

Parameters:

Display	[in] Pointer to EGL native display instance created by **fbGetDisplay**.

fbCreateWindow

Description:
This function is used to create a window for the framebuffer platform with the specified position and size. If width/height is 0, it will use the display width/height as its value.

Note: When either window X + width or the Y + height is larger than the display's width or height respectively, the API will reduce the window size to force the whole window inside the display screen limits. To avoid reducing the window size in this scenario, users can set a value of "1" to the environment variable **FB_IGNORE_DISPLAY_SIZE**.

Syntax:

```
EGLNativeWindowType
fbCreateWindow (
      EGLNativeDisplayType   Display,
      int                    X,
      int                    Y,
      int                    Width,
      int                    Height
);
```

Parameters:

Display	[in] Pointer to EGL native display instance created by **fbGetDisplay**.
X	[in] Specifies the initial horizontal position of the window.
Y	[in] Specifies the initial vertical position of the window.
Width	[in] Specifies the width of the window.
Height	[in] Specifies the height of the window in device units.

Return Value:
The function returns a pointer to the EGL native window instance if successful; otherwise, it returns a NULL pointer.

fbGetWindowGeometry

Description:
This function is used to get window position and size information.

Syntax:

```
void
fbGetWindowGeometry (
        EGLNativeWindowType     Window,
        int *                   X,
        int *                   Y,
        int *                   Width,
        int *                   Height
);
```

Parameters:

Window	[in] Pointer to EGL native window instance created by **fbCreateWindow**.
X	[out] Pointer that receives the horizontal position value of the window.
Y	[out] Pointer that receives the vertical position value of the window.
Width	[out] Pointer that receives the width value of the window.
Height	[out] Pointer that receives the height value of the window.

fbGetWindowInfo

Description:
This function is used to get window position and size and address information.

Syntax:

```
void
fbGetWindowInfo (
        EGLNativeWindowType     Window,
        int *                   X,
        int *                   Y,
        int *                   Width,
        int *                   Height
        int *                   BitsPerPixel,
        unsigned int *          Offset
);
```

Parameters:

Window	[in] A pointer to the EGL native window instance created by **fbCreateWindow**.
X	[out] A pointer to the location that contains the horizontal position value of the window.
Y	[out] A pointer to the location that contains the vertical position value of the window.
Width	[out] A pointer to the location that contains the width of the window.
Height	[out] A pointer to the location that contains the height of the window.
BitsPerPixel	[out] A pointer to the location that contains the pixel depth of the window.
Offset	[out] A pointer to the location that contains the offset of the window.

fbDestroyWindow

Description:
This function is used to destroy a window.

Syntax:

```
void
fbDestroyWindow (
        EGLNativeWindowType     Window
);
```

Parameters:

Window	[in] Pointer to EGL native window instance created by **fbCreateWindow**.

fbCreatePixmap

Description:
This function is used to create a pixmap of a specific size on the specified framebuffer device. If either the width or height is 0, the function will fail to create a pixmap and return NULL.

Syntax:

```
EGLNativePixmapType
fbCreatePixmap (
        EGLNativeDisplayType    Display,
        int                     Width,
        int                     Height
);
```

Parameters:

Display	[in] Pointer to the EGL native display instance created by **fbGetDisplay**.
Width	[in] Specifies the width of the pixmap.
Height	[in] Specifies the height of the pixmap.

Return Value:
The function returns a pointer to the EGL native pixmap instance if successful; otherwise, it returns a NULL pointer.

fbCreatePixmapWithBpp

Description:
This function is used to create a pixmap of a specific size and bit depth on the specified framebuffer device. If either the width or height is 0, the function will fail to create a pixmap and return NULL.

Syntax:

```
EGLNativePixmapType
fbCreatePixmapWithBpp (
        EGLNativeDisplayType    Display,
        int                     Width,
        int                     Height
        int                     BitsPerPixel
);
```

Parameters:

Display	[in]A pointer to the EGL native display instance created by **fbGetDisplay**.
Width	[in] Specifies the width of the pixmap.
Height	[in] Specifies the height of the pixmap.
BitsPerPixel	[in] Specifies the bit depth of the pixmap.

Return Value:
The function returns a pointer to the EGL native pixmap instance if successful; otherwise, it returns a NULL pointer.

fbGetPixmapGeometry

Description:
This function is used to get pixmap size information.

Syntax:

```
void
fbGetPixmapGeometry (
        EGLNativePixmapType     Pixmap,
        int *                   Width,
        int *                   Height
);
```

Parameters:

Pixmap	[in] Pointer to the EGL native pixmap instance created by **fbCreatePixmap**.
Width	[out] Pointer that receives a width value for pixmap.
Height	[out] Pointer that receives a height value for pixmap.

fbGetPixmapInfo

Description:
This function is used to get pixmap size and depth information.
Syntax:

```
void
fbGetPixmapInfo (
        EGLNativePixmapType     Pixmap,
        int *                   Width,
        int *                   Height
        int *                   BitsPerPixel
        int *                   Stride,
        void **                 Bits
);
```

Parameters:

Pixmap	[in] A pointer to the EGL native pixmap instance created by **fbCreatePixmap**.
Width	[out] A pointer to the location that contains a width value for pixmap.
Height	[out] A pointer to the location that contains a height value for pixmap.
BitsPerPixel	[out] A pointer to the location that contains the pixel depth of the pixmap.
Stride	[out] A pointer to the location that contains the stride of the pixmap.
Bits	[out] A pointer to the location that contains the bit address of the pixmap.

fbDestroyPixmap

Description:
This function is used to destroy a pixmap.

Syntax:

```
void
fbDestroyPixmap (
        EGLNativePixmapType     Pixmap
);
```

Parameters:

Pixmap	[in] Pointer to the EGL native pixmap instance created by **fbCreatePixmap**.

4. Revision History

This section describes top level differences between document revisions:

Version	Date	Driver Version	Notes
1.2	2013-07-22	4.6.9-p12	• New driver version included.
1.1	2013-03-04	4.6.9-p9	• Added fbGetDisplayInfo, fbGetWindowInfo, and fbGetPixmapInfo • Updated fbCreatePixmapWithBpp
1.0	2012-10-04	4.6.9-p6	Initial release

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Document Number: IMX6FBAPI
Rev. L3.0.35_4.1.0
09/2013