

FA92/VA92 Non-OS BLT Library Reference Guide V1.0

Publication Release Date: May 2013

Support Chips:
W55FA Series

Support Platforms:
Non-OS

The information in this document is subject to change without notice.

The Nuvoton Technology Corp. shall not be liable for technical or editorial errors or omissions contained herein; nor for incidental or consequential damages resulting from the furnishing, performance, or use of this material.

This documentation may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from the Nuvoton Technology Corp.

Nuvoton Technology Corp. All rights reserved.

Table of Contents

1. Introduction.....	5
1.1. Feature	5
1.2. Pixel Mapping.....	5
1.3. Transformation Matrix	6
Scaling	6
Rotation	7
Shearing.....	7
1.4. Color Transformation	7
2. API.....	9
2.1. Data Structure	9
E_BLT_INT_TYPE	9
E_DRVBLT_FILLOP	9
E_DRVBLT_REVEAL_ALPHA	9
E_DRVBLT_TRANSFORM_FLAG	9
E_DRVBLT_BMPixel_FORMAT	10
E_DRVBLT_DISPLAY_FORMAT	10
E_DRVBLT_FILL_STYLE	11
E_DRVBLT_PALETTE_ORDER	11
S_DRVBLT_MATRIX	11
S_DRVBLT_ARGB16.....	11
S_DRVBLT_ARGB8.....	12
S_DRVBLT_SRC_IMAGE	12
S_DRVBLT_DEST_FB	12
2.2. Function	13
bltOpen	13
bltClose.....	13
bltSetTransformMatrix	13
bltGetTransformMatrix.....	14
bltSetSrcFormat	14
bltGetSrcFormat	14
bltSetDisplayFormat	15
bltGetDisplayFormat	15
bltEnableInt	15
bltDisableInt	16
bltIsIntEnabled	16
bltPollInt.....	16
bltInstallCallback.....	17

bltSetColorMultiplier.....	17
bltGetColorMultiplier.....	17
bltSetColorOffset.....	18
bltGetColorOffset.....	18
bltSetSrcImage.....	18
bltSetDestFrameBuf.....	19
bltSetARGBFillColor.....	19
bltGetARGBFillColor.....	19
bltGetBusyStatus.....	20
bltSetFillAlpha.....	20
bltGetFillAlpha.....	21
bltSetTransformFlag.....	21
bltGetTransformFlag.....	21
bltSetPaletteEndian.....	22
bltGetPaletteEndian.....	22
bltSetColorPalette.....	22
bltSetFillOP.....	23
bltGetFillOP.....	23
bltSetFillStyle.....	23
bltGetFillStyle.....	24
bltSetRevealAlpha.....	24
bltGetRevealAlpha.....	24
bltTrigger.....	25
bltSetRGB565TransparentColor.....	25
bltGetRGB565TransparentColor.....	25
bltSetRGB565TransparentCtl.....	26
bltGetRGB565TransparentCtl.....	26
bltFlush.....	26
2.3. Error Code Table.....	27
3. Revision History.....	28

1. Introduction

This document is written for user applications which want to make use of BLT through provided API.

1.1. Feature

- Fill operation.
 - Fill color with alpha channel
- Blit operation
 - Transformation effects (Scaling, Rotation, Shearing, etc.) through 2x2 inverse transformation matrix.
 - Bitmap smoothing in bi-linear algorithm.
 - Tiling mode (for inversely mapped source pixels lying outside the boundaries of the source image)
 - ◆ No drawing
 - ◆ Clip to edge (closest edge pixel of the source image)
 - ◆ Repeat (source image repeated indefinitely in all directions)
 - Color transformation as defined in Adobe Flash
 - RGB565 color key
- Source format for Blit operation
 - ARGB8888
 - RGB565
 - Palette index with color ARGB8888
 - ◆ 1-bit, 2-bit, 4-bit, and 8-bit palette index
 - ◆ Endianness of palette index
- Destination format for Fill/Blit operation
 - ARGB8888
 - RGB555
 - RGB565

1.2. Pixel Mapping

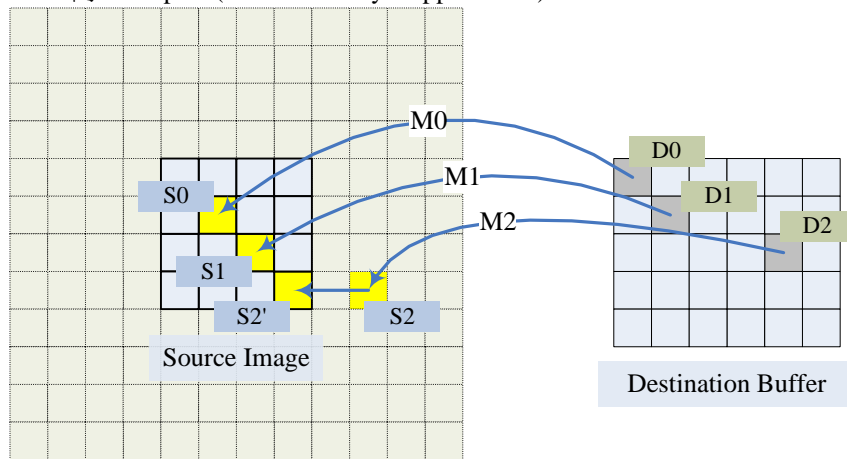
To use blit operation, think of pixel mapping in the inverse direction, that is, from destination to source. Below is an example which demos how pixels are inversely mapped with identify transformation matrix.

1. The transformation matrix for Blit operation must be inverse. That is, matrices in [bltSetTransformMatrix](#) and [bltGetTransformMatrix](#) must be inverse. Elements a, b, c, and d in [S_DRVBLT_MATRIX](#) must fill as below.

$$\begin{pmatrix} x_d \\ y_d \end{pmatrix} = \begin{pmatrix} s & t \\ u & v \end{pmatrix} \begin{pmatrix} x_s \\ y_s \end{pmatrix}$$

$$\begin{pmatrix} s & t \\ u & v \end{pmatrix}^{-1} \begin{pmatrix} x_d \\ y_d \end{pmatrix} = \begin{pmatrix} x_s \\ y_s \end{pmatrix}$$

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} s & t \\ u & v \end{pmatrix}^{-1}$$
2. In M0, D0 (origin pixel of destination buffer) is inversely mapped to S0, which is specified in (i32XOffset, i32YOffset) of [S_DRVBLT_SRC_IMAGE](#), and needn't be the origin pixel of the source image.
3. In M1, D1 is inversely mapped to S1, which lies within the boundaries of the source image.
4. In M2, D2 is inversely mapped to S2, which lies outside the boundaries of the source image. Dependent on tiling mode specified in [E_DRVBLT_FILL_STYLE](#), there are 3 different rendering results:
 - 甲、No drawing (D2 is not drawn).
 - 乙、Clip to edge (D2 is inversely mapped to S2').
 - 丙、Repeat (D2 is inversely mapped to S2).



1.3. Transformation Matrix

In Blit operation, transformation effects, such as Scaling, Rotation, Shearing, etc. can be achieved through a (inverse) transformation matrix. Note as mentioned above, example matrices here are forward, but they must be inverse in Blit setup.

Scaling

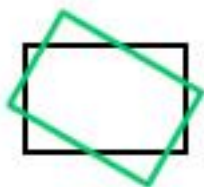
Resize the image by multiplying the location of each pixel by s_x on the x axis and s_y on the y axis.



$$\begin{pmatrix} x_d \\ y_d \end{pmatrix} = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} \begin{pmatrix} x_s \\ y_s \end{pmatrix}$$

Rotation

Rotate the image by an angle θ .



$$\begin{pmatrix} x_d \\ y_d \end{pmatrix} = \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x_s \\ y_s \end{pmatrix}$$

Shearing

Slide the image in a direction parallel to the x axis.



$$\begin{pmatrix} x_d \\ y_d \end{pmatrix} = \begin{pmatrix} 1 & k \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_s \\ y_s \end{pmatrix}$$

1.4. Color Transformation

In Blit operation, user application can decide to apply color transformation or not, which is defined by Adobe Flash and has the following formula. Besides, user application can further decide to apply the alpha channel only.

New alpha value = (old alpha value * alphaMultiplier) + alphaOffset

New red value = (old red value * redMultiplier) + redOffset

New green value = (old green value * greenMultiplier) + greenOffset
New blue value = (old blue value * blueMultiplier) + blueOffset

2. API

2.1. Data Structure

E_BLT_INT_TYPE

Interrupt type.

Name	Value	Description
<i>BLT_INT_CMPLT</i>	1	Fill/Blit operation completed

E_DRVBLT_FILLOP

Fill or Blit operation.

Name	Value	Description
<i>eDRVBLT_DISABLE</i>	0	Blit operation
<i>eDRVBLT_ENABLE</i>	1	Fill operation

E_DRVBLT_REVEAL_ALPHA

Premultiplied alpha or not for source format of ARGB8888

Name	Value	Description
<i>eDRVBLT_EFFECTIVE</i>	0	Premultiplied alpha
<i>eDRVBLT_NO_EFFECTIVE</i>	1	Non-premultiplied alpha

E_DRVBLT_TRANSFORM_FLAG

Transform flags for Blit operation.

Color transformation formula applied when *eDRVBLT_HASCOLORTTRANSFORM* specified:

New alpha value = (old alpha value * alphaMultiplier) + alphaOffset

New red value = (old red value * redMultiplier) + redOffset

New green value = (old green value * greenMultiplier) + greenOffset

New blue value = (old blue value * blueMultiplier) + blueOffset

Alpha-only color transformation formula applied when both eDRVBLT_HASCOLORTRANSFORM and eDRVBLT_HASALPHAONLY specified:

New alpha value = (old alpha value * alphaMultiplier) + alphaOffset

Name	Value	Description
<i>eDRVBLT_NONTRANSPARENCYE</i>	0	No per-pixel transparency in the source.
<i>eDRVBLT_HASTRANSPARENCY</i>	1	Has per-pixel transparency in the source.
<i>eDRVBLT_HASCOLORTRANSFORM</i>	2	Apply color transformation formula.
<i>eDRVBLT_HASALPHAONLY</i>	4	If color transformation enabled, just apply the alpha-only formula.

E_DRVBLT_BMPIXEL_FORMAT

Source format for Blit operation.

If eDRVBLT_SRC_ARGB8888/palette index, source/palette color can be RGB888 or ARGB8888 dependent on [E_DRVBLT_TRANSFORM_FLAG](#).

Name	Value	Description
<i>eDRVBLT_SRC_ARGB8888</i>	1	RGB888/ARGB8888
<i>eDRVBLT_SRC_RGB565</i>	2	RGB565
<i>eDRVBLT_SRC_1BPP</i>	4	1-bit palette index
<i>eDRVBLT_SRC_2BPP</i>	8	2-bit palette index
<i>eDRVBLT_SRC_4BPP</i>	16	4-bit palette index
<i>eDRVBLT_SRC_8BPP</i>	32	8-bit palette index

E_DRVBLT_DISPLAY_FORMAT

Destination format for Fill/Blit operation.

Name	Value	Description
<i>eDRVBLT_DEST_ARGB8888</i>	1	ARGB8888
<i>eDRVBLT_DEST_RGB565</i>	2	RGB565
<i>eDRVBLT_DEST_RGB555</i>	4	RGB555

E_DRVBLT_FILL_STYLE

Other flags for Blit operation.

eDRVBLT_CLIP_TO_EDGE/eDRVBLT_NONE_FIL specify how to behave when reverse mapping doesn't fall in the range of source bitmap.

Name	Value	Description
<i>eDRVBLT_CLIP_TO_EDGE</i>	1	The bitmap should be clipped to its edges, otherwise a repeating texture.
<i>eDRVBLT_NOTSMOOTH</i>	2	The bitmap should not be smoothed
<i>eDRVBLT_NONE_FILL</i>	4	Neither clip to edge nor repeating texture

E_DRVBLT_PALETTE_ORDER

Palette index in big-endian or little-endian.

Name	Value	Description
<i>eDRVBLT_BIG_ENDIAN</i>	0	Palette index in big endian
<i>eDRVBLT_LITTLE_ENDIAN</i>	1	Palette index in little endian

S_DRVBLT_MATRIX

Transformation matrix used in inverse mapping..

Name	Type	Description
<i>a</i>	INT32	
<i>b</i>	INT32	
<i>c</i>	INT32	
<i>d</i>	INT32	

S_DRVBLT_ARGB16

Multiplier/offset of A, R, G, and B channels used in color transformation.

Name	Type	Description
<i>i16Blue</i>	INT16	Color multiplier/offset of blue channel
<i>i16Green</i>	INT16	Color multiplier/offset of green channel
<i>i16Red</i>	INT16	Color multiplier/offset of red channel
<i>i16Alpha</i>	INT16	Color multiplier/offset of alpha channel

S_DRVBLT_ARGB8

ARGB8888 color

Name	Type	Description
<i>u8Blue</i>	UINT8	Value of blue channel
<i>u8Green</i>	UINT8	Value of green channel
<i>u8Red</i>	UINT8	Value of red channel
<i>u8Alpha</i>	UINT8	Value of alpha channel

S_DRVBLT_SRC_IMAGE

Source image.

Name	Type	Description
<i>u32SrcImageAddr</i>	UINT32	Source image start address
<i>i32Stride</i>	INT32	Source image's stride in bytes
<i>i32XOffset</i>	INT32	X offset into the source to start rendering from
<i>i32YOffset</i>	INT32	Y offset into the source to start rendering from
<i>i16Width</i>	INT16	Source image's width in pixels
<i>i16Height</i>	INT16	Source image's height in pixels

S_DRVBLT_DEST_FB

Destination buffer.

Name	Type	Description
<i>u32FrameBufAddr</i>	UINT32	Destination buffer address to start rendering to
<i>i32XOffset</i>	INT32	No use
<i>i32YOffset</i>	INT32	No use
<i>i32Stride</i>	INT32	Destination buffer's stride in bytes
<i>i16Width</i>	INT16	Destination buffer's width in pixels
<i>i16Height</i>	INT16	Destination buffer's height in pixels

2.2. Function

bltOpen

Synopsis

```
ERRCODE bltOpen(void);
```

Description

Initialize BLT and install interrupt service routine.

Parameter

None

Return Value

E_SUCCESS	Success
-----------	---------

bltClose

Synopsis

```
void bltClose(void);
```

Description

Tear down BLT.

Parameter

None

Return Value

None

bltSetTransformMatrix

Synopsis

```
void bltSetTransformMatrix(S_DRVBLT_MATRIX sMatrix);
```

Description

Set up inverse transformation matrix.

Parameter

sMatrix	Transformation matrix as defined in
S_DRVBLT_MATRIX .	

Return Value

None

bltGetTransformMatrix

Synopsis

```
void bltGetTransformMatrix(S_DRVBLT_MATRIX *psMatrix);
```

Description

Retrieve inverse transformation matrix which has set up.

Parameter

psMatrix	User-prepared buffer to save read-back transformation matrix as defined in S_DRVBLT_MATRIX .
----------	--

Return Value

None

bltSetSrcFormat

Synopsis

```
ERRCODE bltSetSrcFormat (E_DRVBLT_BMPIXEL_FORMAT eSrcFmt);
```

Description

Set up source format.

Parameter

eSrcFmt	Source format as defined in E_DRVBLT_BMPixel_FORMAT .
---------	---

Return Value

E_SUCCESS	Success
ERR_BLT_INVALID_SRCFMT	Invalid source format

bltGetSrcFormat

Synopsis

```
E_DRVBLT_BMPixel_Format bltGetSrcFormat(void);
```

Description

Retrieve source format which has set up.

Parameter

None

Return Value

Source format as defined in [E_DRVBLT_BMPIXEL_FORMAT](#).

bltSetDisplayFormat

Synopsis

```
ERRCODE bltSetDisplayFormat(E_DRVBLT_DISPLAY_FORMAT eDisplayFmt);
```

Description

Set up destination format.

Parameter

eDisplayFmt	Destination format defined in E_DRVBLT_DISPLAY_FORMAT .
-------------	---

Return Value

E_SUCCESS	Success
ERR_BLT_INVALID_DSTFMT	Invalid destination format

bltGetDisplayFormat

Synopsis

```
E_DRVBLT_DISPLAY_FORMAT bltGetDisplayFormat(void);
```

Description

Retrieve destination format which has set up.

Parameter

None

Return Value

Destination format as defined in [E_DRVBLT_DISPLAY_FORMAT](#).

bltEnableInt

Synopsis

```
void bltEnableInt(E_BLT_INT_TYPE eIntType);
```

Description

Enable specified interrupt type.

Parameter

eIntType	Interrupt type as defined in E_BLT_INT_TYPE .
----------	---

Return Value

None

bltDisableInt

Synopsis

```
void bltDisableInt(E_BLT_INT_TYPE eIntType);
```

Description

Disable specified interrupt type.

Parameter

eIntType	Interrupt type as defined in E_BLT_INT_TYPE .
----------	---

Return Value

None

bltIsIntEnabled

Synopsis

```
BOOL bltIsIntEnabled (E_BLT_INT_TYPE eIntType);
```

Description

Query if the specified interrupt type is enabled.

Parameter

eIntType	Interrupt type as defined in E_BLT_INT_TYPE .
----------	---

Return Value

TRUE	Specified interrupt enabled
FALSE	Specified interrupt disabled

bltPollInt

Synopsis

```
BOOL bltPollInt(E_BLT_INT_TYPE eIntType);
```

Description

Query interrupt status of the specified interrupt type.

Parameter

eIntType	Interrupt type as defined in E_BLT_INT_TYPE .
----------	---

Return Value

TRUE	Specified interrupt type active.
FALSE	Specified interrupt type inactive.

bltInstallCallback

Synopsis

```
void bltInstallCallback (E_BLT_INT_TYPE eIntType, PFN_BLT_CALLBACK pfnCallback,
PFN_BLT_CALLBACK* pfnOldCallback);
```

Description

Install callback function invocated on interrupt generated.

Parameter

eIntType	Interrupt type as defined in E_BLT_INT_TYPE .
pfnCallback	New callback function to install. NULL to uninstall.
pfnOldCallback function.	User-prepared buffer to save previously installed callback

Return Value

None

bltSetColorMultiplier

Synopsis

```
void bltSetColorMultiplier(S_DRVBLT_ARGB16 sARGB16);
```

Description

Set up color multipliers of A, R, G, and B channels for color transformation.

Parameter

sARGB16 S_DRVBLT_ARGB16 .	Color multipliers of A, R, G, and B channels as defined in
--	--

Return Value

None

bltGetColorMultiplier

Synopsis

```
void bltGetColorMultiplier(S_DRVBLT_ARGB16* psARGB16);
```

Description

Retrieve color multipliers of A, R, G, and B channels which has set up.

Parameter

psARGB16 User-prepared buffer to save color multipliers of A, R, G, and B channels as defined in [S_DRVBLT_ARGB16](#).

Return Value

None

bltSetColorOffset

Synopsis

```
void bltSetColorOffset(S_DRVBLT_ARGB16 sARGB16);
```

Description

Set up color offsets of A, R, G, and B channels for color transformation.

Parameter

sARGB16 Color offsets of A, R, G, and B channels as defined in [S_DRVBLT_ARGB16](#).

Return Value

None

bltGetColorOffset

Synopsis

```
void bltGetColorOffset(S_DRVBLT_ARGB16* psARGB16);
```

Description

Retrieve color offsets of A, R, G, and B channels which has set up.

Parameter

psARGB16 User-prepared buffer to save color offsets of A, R, G, and B channels as defined in [S_DRVBLT_ARGB16](#).

Return Value

None

bltSetSrcImage

Synopsis

```
void bltSetSrcImage(S_DRVBLT_SRC_IMAGE sSrcImage);
```

Description

Set up source image..

Parameter

sSrcImage

Source image as defined in `S_DRVBLT_SRC_IMAGE`.

Return Value

None

bltSetDestFrameBuf

Synopsis

```
void bltSetDestFrameBuf(S_DRVBLT_DEST_FB sFrameBuf);
```

Description

Set up destination buffer..

Parameter

sFrameBuf

Destination buffer as defined in [S DRVBLT DEST FB](#).

Return Value

None

bltSetARGBFillColor

Synopsis

```
void bltSetARGBFillColor(S_DRVBLT_ARGB8 sARGB8);
```

Description

Set up fill color for Fill operation, which can be ARGB8888 or RGB888 dependent on [bltSetFillAlpha](#).

Parameter

sRGB8

Fill color as defined in [S_DRVBLT_ARGB8](#).

Return Value

None

Note

If ARGB8888, it must be in non-premultiplied alpha format.

bltGetARGBFillColor

Synopsis

```
void bltGetARGBFillColor(S_DRVBLT_ARGB8* psARGB8 );
```

Description

Retrieve ARGB8888 color for Fill operation which has set up.

Parameter

psARGB8	User-prepared buffer to save read-back ARGB8888 color
for Fill operation.	

Return Value

None

bltGetBusyStatus

Synopsis

```
BOOL bltGetBusyStatus(void);
```

Description

Query if Fill/Blit operation is busy.

Parameter

None

Return Value

TRUE	Busy
FALSE	Free

bltSetFillAlpha

Synopsis

```
void bltSetFillAlpha(BOOL bEnable);
```

Description

Set up whether or not fill color's alpha channel is in effect.

Parameter

bEnable	
TRUE	Fill color is ARGB8888
FALSE	Fill color is RGB888

Return Value

None

bltGetFillAlpha

Synopsis

```
BOOL bltGetFillAlpha(void);
```

Description

Retrieve whether or not fill color's alpha channel is in effect which has set up.

Parameter

None

Return Value

TRUE	Fill color is ARGB8888.
FALSE	Fill color is RGB888

bltSetTransformFlag

Synopsis

```
void bltSetTransformFlag(UINT32 u32TransFlag);
```

Description

Set up transform flag.

Parameter

U32TransFlag	Transform flag as defined in E_DRVBLT_TRANSFORM_FLAG .
--------------	--

Return Value

None

bltGetTransformFlag

Synopsis

```
UINT32 bltGetTransformFlag(void);
```

Description

Retrieve transform flag which has set up.

Parameter

None.

Return Value

Transform flag as defined in [E_DRVBLT_TRANSFORM_FLAG](#).

bltSetPaletteEndian

Synopsis

```
void bltSetPaletteEndian(E_DRVBLT_PALETTE_ORDER eEndian);
```

Description

Set up endianness of palette index..

Parameter

eEndian Endianness of palette index as defined in [E_DRVBLT_PALETTE_ORDER](#).

Return Value

None

bltGetPaletteEndian

Synopsis

```
E_DRVBLT_PALETTE_ORDER bltGetPaletteEndian(void);
```

Description

Retrieve endianness of palette index which has set up.

Parameter

None

Return Value

Endianness of palette index as defined in [E_DRVBLT_PALETTE_ORDER](#).

bltSetColorPalette

Synopsis

```
void bltSetColorPalette(UINT32 u32PaletteInx, UINT32 u32Num, S_DRVBLT_ARGB8 *psARGB);
```

Description

Set up palette's colors.

Parameter

u32PaletteInx	Index of palette to start to set up
u32Num	Number of colors to set up
psARGB	ARGB8888 colors

Return Value

None

bltSetFillOP

Synopsis

```
void bltSetFillOP(E_DRVBLT_FILLOP eOP);
```

Description

Set up operation to be Fill or Blit.

Parameter

eOP	Operation as defined in E_DRVBLT_FILLOP .
-----	---

Return Value

None

bltGetFillOP

Synopsis

```

BOOL bltGetFillOP(void);

```

Description

Retrieve operation which has set up..

Parameter

None

Return Value

TRUE Fill operation.

FALSE Blit operation

bltSetFillStyle

Synopsis

```
void bltSetFillStyle(E_DRVBLT_FILL_STYLE eStyle);
```

Description

Set up other flags for Blit operation.

Parameter

eStyle	Other flags as defined in E_DRVBLT_FILL_STYLE .
--------	---

Return Value

None

bltGetFillStyle

Synopsis

```
E_DRVBLT_FILL_STYLE bltGetFillStyle(void);
```

Description

Retrieve other flags for Blit operation which has set up.

Parameter

None

Return Value

Other flags as defined in [E_DRVBLT_FILL_STYLE](#).

bltSetRevealAlpha

Synopsis

```
void bltSetRevealAlpha(E_DRVBLT_REVEAL_ALPHA eAlpha);
```

Description

Set up premultiplied alpha or not for source format of ARGB8888

Parameter

eAlpha Premultiplied alpha or not as specified in
[E_DRVBLT_REVEAL_ALPHA](#)

Return Value

None

bltGetRevealAlpha

Synopsis

```
BOOL bltGetRevealAlpha(void);
```

Description

Retrieve premultiplied alpha or not for source format of ARGB8888.

Parameter

None

Return Value

Premultiplied alpha or not as specified in [E_DRVBLT_REVEAL_ALPHA](#)

bltTrigger

Synopsis

void bltTrigger(void);

Description

Start Fill/Blit operation..

Parameter

None

Return Value

None

bltSetRGB565TransparentColor

Synopsis

void bltSetRGB565TransparentColor(UINT16 u16RGB565);

Description

Set up transparent color for source format of RGB565 for color key enabled

Parameter

u16RGB565 RGB565 to be transparent color

Return Value

None

bltGetRGB565TransparentColor

Synopsis

UINT16 bltGetRGB565TransparentColor(void);

Description

Retrieve transparent color which has set up..

Parameter

None

Return Value

RGB565 to be transparent color

bltSetRGB565TransparentCtl

Synopsis

```
void bltSetRGB565TransparentCtl(BOOL bEnable);
```

Description

Enable color key or not.

Parameter

bEnable

TRUE	Enable color key
FALSE	Disable color key

Return Value

None

bltGetRGB565TransparentCtl

Synopsis

```
BOOL bltGetRGB565TransparentCtl(void);
```

Description

Retrieve color key enabled or not.

Parameter

None

Return Value

TRUE	Color key enabled
FALSE	Color key disabled

bltFlush

Synopsis

```
void bltFlush(void);
```

Description

Wait for Fill/Blit operation to complete.

Parameter

None

Return Value

None

2.3. Error Code Table

Code Name	Value	Description
Successful	0	Success
ERR_BLT_INVALID_INT	BLT_ERR_ID 0x01	Invalid interrupt type
ERR_BLT_INVALID_SRCFMT	BLT_ERR_ID 0x02	Invalid source format
ERR_BLT_INVALID_DSTFMT	BLT_ERR_ID 0x01	Invalid destination format

3. Revision History

Version	Date	Description
V1	May 08, 2013	• Created

Important Notice

Nuvoton products are not designed, intended, authorized or warranted for use as components in equipment or systems intended for surgical implantation, atomic energy control instruments, aircraft or spacecraft instruments, transportation instruments, traffic signal instruments, combustion control instruments, or for any other applications intended to support or sustain life. Furthermore, Nuvoton products are not intended for applications whereby failure could result or lead to personal injury, death or severe property or environmental damage.

Nuvoton customers using or selling these products for such applications do so at their own risk and agree to fully indemnify Nuvoton for any damages resulting from their improper use or sales.