

# display-dma2d

## *User Manual*



**MICROEJ<sup>®</sup>**

Reference:	TLT-XXX-MAN-display-dma2d-display-dma2d
Version:	3.0.0
Revision:	XXX

---

## Confidentiality & Intellectual Property

All rights reserved. Information, technical data and tutorials contained in this document are confidential and proprietary under copyright Law of Industrial Smart Software Technology (IS2T S.A.) operating under the brand name MicroEJ®. Without written permission from IS2T S.A., *copying or sending parts of the document or the entire document by any means to third parties is not permitted*. Granted authorizations for using parts of the document or the entire document do not mean IS2T S.A. gives public full access rights.

The information contained herein is not warranted to be error-free. IS2T® and MicroEJ® and all relative logos are trademarks or registered trademarks of IS2T S.A. in France and other Countries.

Java™ is Sun Microsystems' trademark for a technology for developing application software and deploying it in cross-platform, networked environments. When it is used in this documentation without adding the ™ symbol, it includes implementations of the technology by companies other than Sun.

Java™, all Java-based marks and all related logos are trademarks or registered trademarks of Sun Microsystems Inc, in the United States and other Countries.

Other trademarks are proprietary of their authors.

---

---

# Table of Contents

1. Data Structure Documentation .....	1
1.1. DRAWING_DMA2D_blending_t struct Reference .....	1
1.1.1. Data Fields .....	1
1.1.2. Field Documentation .....	1
1.2. DRAWING_DMA2D_memcpy struct Reference .....	1
1.2.1. Data Fields .....	1
1.2.2. Field Documentation .....	2
2. File Documentation .....	3
2.1. drawing_dma2d.h File Reference .....	3
2.1.1. Data Structures .....	3
2.1.2. Functions .....	3
2.2. drawing_dma2d.c File Reference .....	4
2.2.1. Data Structures .....	4
2.2.2. Typedefs .....	4
2.2.3. Variables .....	4
2.2.4. Functions .....	4

---

# Chapter 1. Data Structure Documentation

## 1.1. DRAWING\_DMA2D\_blending\_t struct Reference

### 1.1.1. Data Fields

- uint8\_t \* src\_address
- uint8\_t \* dest\_address
- uint32\_t src\_stride
- uint32\_t dest\_stride
- jchar dest\_width
- jchar dest\_height
- jint x\_src
- jint y\_src
- jint width
- jint height
- jint x\_dest
- jint y\_dest
- jint alpha
- uint32\_t src\_dma2d\_format
- uint32\_t src\_bpp

### 1.1.2. Field Documentation

## 1.2. DRAWING\_DMA2D\_memcpy struct Reference

### 1.2.1. Data Fields

- uint8\_t \* src\_address
- uint8\_t \* dest\_address
- uint16\_t width

- `uint16_t` height

## 1.2.2. Field Documentation

---

# Chapter 2. File Documentation

## 2.1. drawing\_dma2d.h File Reference

```
#include <stdint.h>
```

```
#include "LLUI_DISPLAY_IMPL.h"
```

```
#include "sni.h"
```

### 2.1.1. Data Structures

- struct DRAWING\_DMA2D\_memcpy

### 2.1.2. Functions

- void DRAWING\_DMA2D\_initialize ( void \* binary\_semaphore\_handle)
- void DRAWING\_DMA2D\_IRQHandler ( void )
- void DRAWING\_DMA2D\_configure\_memcpy ( uint8\_t \* srcAddr, uint8\_t \* destAddr, uint32\_t xmin, uint32\_t ymin, uint32\_t xmax, uint32\_t ymax, uint32\_t stride, DRAWING\_DMA2D\_memcpy \* memcpy\_data)
- void DRAWING\_DMA2D\_start\_memcpy ( DRAWING\_DMA2D\_memcpy \* memcyp\_data)

## Detailed Description

Use STM32 DMA2D (ChromART) for MicroEJ ui\_drawing.h implementation.

This library provides the implementation of ui\_drawing.h "UI\_DRAWING\_fillRectangle()" and "UI\_DRAWING\_drawImage()" functions. The third feature "memcpy" is useful when a copy from frame buffer to back buffer is required after the call to "LLUI\_DISPLAY\_IMPL\_flush()".

How to use this library:

- Set the define DRAWING\_DMA2D\_BPP to 16, 24 or 32
- Set the define STM32F4XX or STM32F7XX
- Call "DRAWING\_DMA2D\_initialize()" during "LLUI\_DISPLAY\_IMPL\_initialize()"
- Redirect the STM32 DMA2D interrupt routine to "DRAWING\_DMA2D\_IRQHandler()"

- Call "DRAWING\_DMA2D\_configure\_memcpy()" in "LLUI\_DISPLAY\_IMPL\_flush()" before enabling LCD interrupt (optional).
- Call "DRAWING\_DMA2D\_start\_memcpy()" in LCD interrupt (optional).

Author: . MicroEJ Developer Team

Version: . 3.0.0

Date: . 31 August 2022

Definition in file C:/Jenkins/workspace/master77851a12/bsp-lldisplay\_dma2d/target~/ccomponent-Working/bsp/ui/inc/drawing\_dma2d.h

## 2.2. drawing\_dma2d.c File Reference

```
#include "drawing_dma2d.h"
```

```
#include "ui_drawing.h"
```

```
#include "ui_drawing_soft.h"
```

```
#include "bsp_util.h"
```

### 2.2.1. Data Structures

- struct DRAWING\_DMA2D\_blending\_t

### 2.2.2. Typedefs

- typedef void(\* t\_drawing\_notification

### 2.2.3. Variables

- static DMA2D\_HandleTypeDef g\_hdma2d
- static t\_drawing\_notification g\_callback\_notification
- static bool g\_dma2d\_running
- static void \* g\_dma2d\_semaphore

### 2.2.4. Functions

- static void \_drawing\_dma2d\_wait ( void )

- static void \_drawing\_dma2d\_done ( void )
- static uint8\_t \* \_drawing\_dma2d\_adjust\_address ( uint8\_t \* address, uint32\_t x, uint32\_t y, uint32\_t stride, uint32\_t bpp)
- static void \_drawing\_dma2d\_configure\_alpha\_image\_data ( MICROUI\_GraphicsContext \* gc, jint \* alphaAndColor)
- static void \_cleanDCache ( void )
- static bool \_drawing\_dma2d\_is\_image\_compatible\_with\_dma2d ( MICROUI\_GraphicsContext \* gc, MICROUI\_Image \* image, jint x\_src, jint y\_src, jint width, jint height, jint x\_dest, jint y\_dest, jint alpha, DRAWING\_DMA2D\_blending\_t \* dma2d\_blending\_data)
- static void \_drawing\_dma2d\_blending\_configure ( DRAWING\_DMA2D\_blending\_t \* dma2d\_blending\_data)
- static void \_drawing\_dma2d\_blending\_start ( DRAWING\_DMA2D\_blending\_t \* dma2d\_blending\_data)
- static void \_drawing\_dma2d\_overlap\_horizontal ( DRAWING\_DMA2D\_blending\_t \* dma2d\_blending\_data)
- static void \_drawing\_dma2d\_overlap\_vertical ( DRAWING\_DMA2D\_blending\_t \* dma2d\_blending\_data)
- void DRAWING\_DMA2D\_IRQHandler ( void )
- void DRAWING\_DMA2D\_initialize ( void \* binary\_semaphore\_handle)
- void DRAWING\_DMA2D\_configure\_memcpy ( uint8\_t \* srcAddr, uint8\_t \* destAddr, uint32\_t xmin, uint32\_t ymin, uint32\_t xmax, uint32\_t ymax, uint32\_t stride, DRAWING\_DMA2D\_memcpy \* memcpy\_data)
- void DRAWING\_DMA2D\_start\_memcpy ( DRAWING\_DMA2D\_memcpy \* memcpy\_data)
- DRAWING\_Status UI\_DRAWING\_fillRectangle ( MICROUI\_GraphicsContext \* gc, jint x1, jint y1, jint x2, jint y2)
- DRAWING\_Status UI\_DRAWING\_drawImage ( MICROUI\_GraphicsContext \* gc, MICROUI\_Image \* image, jint x\_src, jint y\_src, jint width, jint height, jint x\_dest, jint y\_dest, jint alpha)
- DRAWING\_Status UI\_DRAWING\_drawRegion ( MICROUI\_GraphicsContext \* gc, jint x\_src, jint y\_src, jint width, jint height, jint x\_dest, jint y\_dest, jint alpha)

## Detailed Description

Use STM32 DMA2D (ChromART) for MicroEJ ui\_drawing.h implementation.

Author: . MicroEJ Developer Team

Version: . 3.0.0



Date: . 31 August 2022

Definition in file C:/Jenkins/workspace/master77851a12/bsp-lldisplay\_dma2d/target~/ccomponent-Working/bsp/ui/src/drawing\_dma2d.c