

microui

User Manual



MICROEJ[®]

Reference:	TLT-XXX-MAN-microui-microui
Version:	2.0.0
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. File Documentation	1
1.1. microui_event_decoder.h File Reference	1
1.1.1. Typedefs	1
1.1.2. Functions	1
1.2. microui_event_decoder_conf.h File Reference	1
1.2.1. Macros	2
1.3. microui_heap.h File Reference	2
1.3.1. Functions	2
1.4. LLDW_PAINTER_impl.c File Reference	2
1.4.1. Macros	3
1.4.2. Functions	3
1.5. LLUI_DISPLAY_HEAP_impl.c File Reference	4
1.5.1. Macros	4
1.5.2. Variables	5
1.5.3. Functions	5
1.6. LLUI_INPUT_LOG_impl.c File Reference	5
1.7. LLUI_PAINTER_impl.c File Reference	6
1.7.1. Macros	6
1.7.2. Functions	7
1.8. microui_event_decoder.c File Reference	8

Chapter 1. File Documentation

1.1. microui_event_decoder.h File Reference

```
#include <stdlib.h>
```

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

```
#include <stdbool.h>
```

```
#include "microui_event_decoder_conf.h"
```

1.1.1. Typedefs

- typedef void(* MICROUI_EVENT_DECODER_decode_event_data

1.1.2. Functions

- void MICROUI_EVENT_DECODER_describe_dump_start (void)
- void MICROUI_EVENT_DECODER_describe_dump_past (void)
- void MICROUI_EVENT_DECODER_describe_dump_future (void)
- void MICROUI_EVENT_DECODER_describe_dump_events_objects (void)
- void MICROUI_EVENT_DECODER_describe_dump_end (void)
- void MICROUI_EVENT_DECODER_drop_data (uint32_t data, uint32_t index)
- void MICROUI_EVENT_DECODER_decode_event (uint32_t event, uint32_t index, MICROUI_EVENT_DECODER_decode_event_data * fct_data_decoder)

Detailed Description

Definition in file C:/Jenkins/workspace/masterfb994a57/bsp-llmicroui/target~/ccomponentWorking/bsp/ui/inc/microui_event_decoder.h

1.2. microui_event_decoder_conf.h File Reference

```
#include <stdio.h>
```

```
#include "microui_constants.h"
```

1.2.1. Macros

- #define MICROUIEVENTDECODER_ENABLED
- #define MICROUIEVENTDECODER_EVENTGEN_COMMAND MICROUI_EVENTGEN_COMMANDS
- #define MICROUIEVENTDECODER_EVENTGEN_BUTTONS MICROUI_EVENTGEN_BUTTONS
- #define MICROUIEVENTDECODER_EVENTGEN_TOUCH MICROUI_EVENTGEN_TOUCH
- #define LLUI_DEBUG_TRACE (void)printf

Detailed Description

Definition in file C:/Jenkins/workspace/masterfb994a57/bsp-llmicroui/target~/ccomponentWorking/bsp/ui/inc/microui_event_decoder_conf.h

1.3. microui_heap.h File Reference

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

1.3.1. Functions

- uint32_t MICROUI_HEAP_total_space (void)
- uint32_t MICROUI_HEAP_free_space (void)
- uint32_t MICROUI_HEAP_number_of_allocated_blocks (void)

Detailed Description

Definition in file C:/Jenkins/workspace/masterfb994a57/bsp-llmicroui/target~/ccomponentWorking/bsp/ui/inc/microui_heap.h

1.4. LLDW_PAINTER_impl.c File Reference

```
#include "LLDW_PAINTER_impl.h"
```

```
#include "dw_drawing.h"
```

```
#include "LLUI_DISPLAY.h"
```

1.4.1. Macros

- `#define LOG_DRAW_START LLUI_DISPLAY_logDrawingStart(CONCAT_DEFINES(LOG_DRAW_, fn))`
- `#define LOG_DRAW_END LLUI_DISPLAY_logDrawingEnd(CONCAT_DEFINES(LOG_DRAW_, fn))`
- `#define LOG_DRAW_drawThickFadedPoint 100`
- `#define LOG_DRAW_drawThickFadedLine 101`
- `#define LOG_DRAW_drawThickFadedCircle 102`
- `#define LOG_DRAW_drawThickFadedCircleArc 103`
- `#define LOG_DRAW_drawThickFadedEllipse 104`
- `#define LOG_DRAW_drawThickLine 105`
- `#define LOG_DRAW_drawThickCircle 106`
- `#define LOG_DRAW_drawThickEllipse 107`
- `#define LOG_DRAW_drawThickCircleArc 108`
- `#define LOG_DRAW_drawFlippedImage 200`
- `#define LOG_DRAW_drawRotatedImageNearestNeighbor 201`
- `#define LOG_DRAW_drawRotatedImageBilinear 202`
- `#define LOG_DRAW_drawScaledImageNearestNeighbor 203`
- `#define LOG_DRAW_drawScaledImageBilinear 204`

1.4.2. Functions

- `void LLDW_PAINTER_IMPL_drawThickFadedPoint (MICROUI_GraphicsContext * gc, jint x, jint y, jint thickness, jint fade)`
- `void LLDW_PAINTER_IMPL_drawThickFadedLine (MICROUI_GraphicsContext * gc, jint startX, jint startY, jint endX, jint endY, jint thickness, jint fade, DRAWING_Cap startCap, DRAWING_Cap endCap)`
- `void LLDW_PAINTER_IMPL_drawThickFadedCircle (MICROUI_GraphicsContext * gc, jint x, jint y, jint diameter, jint thickness, jint fade)`
- `void LLDW_PAINTER_IMPL_drawThickFadedCircleArc (MICROUI_GraphicsContext * gc, jint x, jint y, jint diameter, jfloat startAngle, jfloat arcAngle, jint thickness, jint fade, DRAWING_Cap start, DRAWING_Cap end)`

- void LLDW_PAINTER_IMPL_drawThickFadedEllipse (MICROUI_GraphicsContext * gc, jint x, jint y, jint width, jint height, jint thickness, jint fade)
- void LLDW_PAINTER_IMPL_drawThickLine (MICROUI_GraphicsContext * gc, jint startX, jint startY, jint endX, jint endY, jint thickness)
- void LLDW_PAINTER_IMPL_drawThickCircle (MICROUI_GraphicsContext * gc, jint x, jint y, jint diameter, jint thickness)
- void LLDW_PAINTER_IMPL_drawThickEllipse (MICROUI_GraphicsContext * gc, jint x, jint y, jint width, jint height, jint thickness)
- void LLDW_PAINTER_IMPL_drawThickCircleArc (MICROUI_GraphicsContext * gc, jint x, jint y, jint diameter, jfloat startAngle, jfloat arcAngle, jint thickness)
- void LLDW_PAINTER_IMPL_drawFlippedImage (MICROUI_GraphicsContext * gc, MICROUI_Image * img, jint regionX, jint regionY, jint width, jint height, jint x, jint y, DRAWING_Flip transformation, jint alpha)
- void LLDW_PAINTER_IMPL_drawRotatedImageNearestNeighbor (MICROUI_GraphicsContext * gc, MICROUI_Image * img, jint x, jint y, jint rotationX, jint rotationY, jfloat angle, jint alpha)
- void LLDW_PAINTER_IMPL_drawRotatedImageBilinear (MICROUI_GraphicsContext * gc, MICROUI_Image * img, jint x, jint y, jint rotationX, jint rotationY, jfloat angle, jint alpha)
- void LLDW_PAINTER_IMPL_drawScaledImageNearestNeighbor (MICROUI_GraphicsContext * gc, MICROUI_Image * img, jint x, jint y, jfloat factorX, jfloat factorY, jint alpha)
- void LLDW_PAINTER_IMPL_drawScaledImageBilinear (MICROUI_GraphicsContext * gc, MICROUI_Image * img, jint x, jint y, jfloat factorX, jfloat factorY, jint alpha)

Detailed Description

Definition in file C:/Jenkins/workspace/masterfb994a57/bsp-llmicroui/target~/ccomponentWorking/bsp/ui/src/LLDW_PAINTER_impl.c

1.5. LLUI_DISPLAY_HEAP_impl.c File Reference

```
#include "microui_heap.h"
```

```
#include "BESTFIT_ALLOCATOR.h"
```

1.5.1. Macros

- #define BESTFITALLOCATOR_HEADER_SIZE (68)
- #define BESTFITALLOCATOR_BLOCK_SIZE (((uint32_t*)((block)-sizeof(uint32_t))) & 0x7ffffff)

1.5.2. Variables

- static BESTFIT_ALLOCATOR image_heap
- static uint32_t heap_size
- static uint32_t free_space
- static uint32_t allocated_blocks_number

1.5.3. Functions

- uint32_t MICROUI_HEAP_total_space (void)
- uint32_t MICROUI_HEAP_free_space (void)
- uint32_t MICROUI_HEAP_number_of_allocated_blocks (void)
- void LLUI_DISPLAY_IMPL_image_heap_initialize (uint8_t * heap_start, uint8_t * heap_limit)
- uint8_t * LLUI_DISPLAY_IMPL_image_heap_allocate (uint32_t size)
- void LLUI_DISPLAY_IMPL_image_heap_free (uint8_t * block)

Detailed Description

This MicroUI images heap allocator replaces the default allocator embedded in the MicroUI Graphics Engine. It is using a best fit allocator and provides some additional APIs to retrieve the heap information: total space, free space, number of blocks allocated.

See also: . LLUI_DISPLAY_impl.h file comment

Author: . MicroEJ Developer Team

Version: . 2.0.0

Date: . 31 August 2022

Since: . MicroEJ UI Pack 13.1.0

Definition in file C:/Jenkins/workspace/masterfb994a57/bsp-llmicroui/target~/ccomponentWorking/bsp/ui/src/LLUI_DISPLAY_HEAP_impl.c

1.6. LLUI_INPUT_LOG_impl.c File Reference

```
#include <assert.h>
```

```
#include <string.h>
```



```
#include "LLUI_INPUT_impl.h"
```

```
#include "microui_event_decoder.h"
```

Detailed Description

This MicroUI FIFO (queue) logger replaces the default logger embedded in the MicroUI Input Engine. For each queue event, it stores the event's data size. This allows to be able to decode the event when `LLUI_INPUT_dump()` is called.

This logger does not interpret the event: it just recognizes the event's first element and event's data. When an event is detected, the logger calls `microui_event_decoder.h` functions.

See also: . `LLUI_INPUT_impl.h` file comment

Author: . MicroEJ Developer Team

Version: . 2.0.0

Date: . 31 August 2022

Since: . MicroEJ UI Pack 13.1.0

Definition in file `C:/Jenkins/workspace/masterfb994a57/bsp-llmicroui/target~/.ccomponentWorking/bsp/ui/src/LLUI_INPUT_LOG_impl.c`

1.7. LLUI_PAINTER_impl.c File Reference

```
#include "LLUI_PAINTER_impl.h"
```

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

```
#include "LLUI_DISPLAY.h"
```

1.7.1. Macros

- `#define LOG_DRAW_START LLUI_DISPLAY_logDrawingStart(CONCAT_DEFINES(LOG_DRAW_, fn))`
- `#define LOG_DRAW_END LLUI_DISPLAY_logDrawingEnd(CONCAT_DEFINES(LOG_DRAW_, fn))`
- `#define LOG_DRAW_writePixel 1`
- `#define LOG_DRAW_drawLine 2`

- `#define LOG_DRAW_drawHorizontalLine` 3
- `#define LOG_DRAW_drawVerticalLine` 4
- `#define LOG_DRAW_drawRectangle` 5
- `#define LOG_DRAW_fillRectangle` 6
- `#define LOG_DRAW_drawRoundedRectangle` 8
- `#define LOG_DRAW_fillRoundedRectangle` 9
- `#define LOG_DRAW_drawCircleArc` 10
- `#define LOG_DRAW_fillCircleArc` 11
- `#define LOG_DRAW_drawEllipseArc` 12
- `#define LOG_DRAW_fillEllipseArc` 13
- `#define LOG_DRAW_drawEllipse` 14
- `#define LOG_DRAW_fillEllipse` 15
- `#define LOG_DRAW_drawCircle` 16
- `#define LOG_DRAW_fillCircle` 17
- `#define LOG_DRAW_drawARGB` 18
- `#define LOG_DRAW_drawImage` 19

1.7.2. Functions

- `static void _check_bound (jint max, jint * bound, jint * size, jint * origin)`
- `void LLUI_PAINTER_IMPL_writePixel (MICROUI_GraphicsContext * gc, jint x, jint y)`
- `void LLUI_PAINTER_IMPL_drawLine (MICROUI_GraphicsContext * gc, jint startX, jint startY, jint endX, jint endY)`
- `void LLUI_PAINTER_IMPL_drawHorizontalLine (MICROUI_GraphicsContext * gc, jint x, jint y, jint length)`
- `void LLUI_PAINTER_IMPL_drawVerticalLine (MICROUI_GraphicsContext * gc, jint x, jint y, jint length)`
- `void LLUI_PAINTER_IMPL_drawRectangle (MICROUI_GraphicsContext * gc, jint x, jint y, jint width, jint height)`
- `void LLUI_PAINTER_IMPL_fillRectangle (MICROUI_GraphicsContext * gc, jint x, jint y, jint width, jint height)`

- void LLUI_PAINTER_IMPL_drawRoundedRectangle (MICROUI_GraphicsContext * gc, jint x, jint y, jint width, jint height, jint cornerEllipseWidth, jint cornerEllipseHeight)
- void LLUI_PAINTER_IMPL_fillRoundedRectangle (MICROUI_GraphicsContext * gc, jint x, jint y, jint width, jint height, jint cornerEllipseWidth, jint cornerEllipseHeight)
- void LLUI_PAINTER_IMPL_drawCircleArc (MICROUI_GraphicsContext * gc, jint x, jint y, jint diameter, jfloat startAngle, jfloat arcAngle)
- void LLUI_PAINTER_IMPL_drawEllipseArc (MICROUI_GraphicsContext * gc, jint x, jint y, jint width, jint height, jfloat startAngle, jfloat arcAngle)
- void LLUI_PAINTER_IMPL_fillCircleArc (MICROUI_GraphicsContext * gc, jint x, jint y, jint diameter, jfloat startAngle, jfloat arcAngle)
- void LLUI_PAINTER_IMPL_fillEllipseArc (MICROUI_GraphicsContext * gc, jint x, jint y, jint width, jint height, jfloat startAngle, jfloat arcAngle)
- void LLUI_PAINTER_IMPL_drawEllipse (MICROUI_GraphicsContext * gc, jint x, jint y, jint width, jint height)
- void LLUI_PAINTER_IMPL_fillEllipse (MICROUI_GraphicsContext * gc, jint x, jint y, jint width, jint height)
- void LLUI_PAINTER_IMPL_drawCircle (MICROUI_GraphicsContext * gc, jint x, jint y, jint diameter)
- void LLUI_PAINTER_IMPL_fillCircle (MICROUI_GraphicsContext * gc, jint x, jint y, jint diameter)
- void LLUI_PAINTER_IMPL_drawImage (MICROUI_GraphicsContext * gc, MICROUI_Image * img, jint regionX, jint regionY, jint width, jint height, jint x, jint y, jint alpha)

Detailed Description

Definition in file C:/Jenkins/workspace/masterfb994a57/bsp-llmicroui/target~/ccomponentWorking/bsp/ui/src/LLUI_PAINTER_impl.c

1.8. microui_event_decoder.c File Reference

```
#include "microui_event_decoder.h"
```

Detailed Description

This MicroUI Events decoder describes the events to the standard output stream.

See also: . LLUI_INPUT_LOG_impl.c file comment

Author: . MicroEJ Developer Team

Version: . 2.0.0

Date: . 31 August 2022

Since: . MicroEJ UI Pack 13.1.0

Definition in file C:/Jenkins/workspace/masterfb994a57/bsp-llmicroui/target~/ccomponentWorking/bsp/ui/src/microui_event_decoder.c