SSD1306_OLED_Device_Driver

Generated by Doxygen 1.8.13

Contents

Chapter 1

Linux Loadable Kernel Module Driver for SSD1306 OLED.

graphics | datalink | driver

Tested on Linux raspberrypi 5.10.103-v7l+ #1529 SMP Tue Mar 8 12:24:00 GMT 2022 armv7l GNU/Linux (Raspberry Pi Buster.)

Documents generated (by doxygen) at /docs/html/files.html

Demo: Displaying text and the dinosaur from chrome browser.

To compile

```
Install the kernel headers.
$ sudo apt install raspberrypi-kernel-headers

Compile
$ sudo make
```

To run:

```
1. First apply device tree overlay by

    $ sudo make dtoverlay

2. Insert the kernel module

    $ sudo make insmod
```

To check for printk log:

```
    $ dmesg
```

To remove the kernel module:

```
    $ sudo dmesg
```

To generate docs by doxygen

```
$ make doxygen
$ cd /docs/html
```

Kanban - TODO

- [x] release-00: Minimal-viable kernel i2c bus module and simple configruation + fill-screen.
- [x] release-01: Add font / image support to the screen datalink layer.
- [ ] release-02: Add user-space interface through sysfs.
- [ ] release-03: Develop the dinosaur game on this screen.

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 4

Class Documentation

4.1 oled_graphics_params_t Struct Reference

Public Attributes

- uint8_t **line**
- uint8_t **cursor_position**
- uint8_t **font_char_width**

The documentation for this struct was generated from the following file:

- graphics.h

Chapter 5

File Documentation

5.1 datalink.c File Reference

Datalink layer implementation for SSD1306 OLED Driver, I2C-based operations.

```
#include "datalink.h"
#include <linux/i2c.h>
#include <linux/init.h>
#include <linux/module.h>
```

Include dependency graph for datalink.c:

5.2 datalink.h File Reference

Header file for SSD1306 controller interface.

```
#include <linux/delay.h>
#include <linux/i2c.h>
#include <linux/init.h>
#include <linux/module.h>
```

Include dependency graph for datalink.h: This graph shows which files directly or indirectly include this file:

Macros

- #define **SET_MEMORY_ADDRESSING_MODE** 0x20
- #define **SET_DISPLAY_START_LINE** 0x40
- #define **SET_DISPLAY_OFF** 0xAE
- #define **SET_DISPLAY_ON** 0xAF
- #define **SET_ENTIRE_DISPLAY_ON** 0xA4
- #define **SET_DISPLAY_OFFSET** 0xD3
- #define **SET_MUX_RATIO** 0xA8
- #define **SET_DEACTIVATE_SCROLL** 0x2E
- #define **SET_CONTRAST_CONTROL** 0x81
- #define **SET_CHARGE_PUMP** 0x8D
- #define **SET_CHARGE_PUMP_ENABLE** 0x14
- #define **SET_COLUMN_ADDRESS** 0x21
- #define **SET_PAGE_ADDRESS** 0x22
- #define **DONT_CARE** 0x00

Enumerations

- enum eControl_t { **COMMAND_CONTROL**, **DATA_CONTROL** }

  Enum type for ssd1306 function to differentiate whether confirguration is a command type or a data byte.

Functions

- void ssd1306_controller_init (void)

  Initialize ssd1306 OLED controller.
- void ssd1306_write_address (eControl_t control_option, uint8_t address, uint8_t param_len, uint8_t *param)

  Write to ssd1306 register address.

5.2.1 Detailed Description

Header file for SSD1306 controller interface.

5.2.2 Function Documentation

5.2.2.1 ssd1306_controller_init()

```
void ssd1306_controller_init (
            void  )
```

Initialize ssd1306 OLED controller.

Parameters

None.	

Returns

None.

Initialize ssd1306 OLED controller.

Parameters

None.	

Returns

None.

Note

Using anonymous array to pass single parameters.

5.2.2.2 ssd1306_write_address()

```
void ssd1306_write_address (
            eControl_t control_option,
            uint8_t address,
            uint8_t param_len,
            uint8_t * p_param )
```

Write to ssd1306 register address.

Parameters

control_option	DATA_CONTROL indicates to transmit data, COMMAND_CONTROL indicates to transmit command.
address	The register address to write param to.
param_len	Length of parameter if there is any.
p_param	Pointer to parameter to be written.

5.3 driver.c File Reference

This file implements the necessary i2c_client probe and remove callbacks on the SSD1306 I2C bus device driver. On top of driver.c, display configurations and initialization are implemented in datalink.c. On top of datalink, OLED printing / graphics are implemented in graphics.c.

```
#include "datalink.h"
#include "graphics.h"
#include <linux/delay.h>
#include <linux/i2c.h>
#include <linux/module.h>
```

Include dependency graph for driver.c:

Functions

- **MODULE_LICENSE** ("GPL")
- **MODULE_AUTHOR** ("Luyao Han")
- **MODULE_DESCRIPTION** ("Linux kernel module driver for ssd1306 oled display")
- static int driver_on_probe (struct i2c_client ∗client, const struct i2c_device_id ∗device_id)

  Callback function on probing (driver-device binding) of the device driver.
- static int driver_on_remove (struct i2c_client ∗client)

  Callback function on the removal of the device driver.
- MODULE_DEVICE_TABLE (of, driver_id)

  This macro describes which devices each specific driver can support. At compilation time, the build process extracts this information out of the driver and builds a table.
- **MODULE_DEVICE_TABLE** (i2c, driver_device_id)
- **module_i2c_driver** (i2c_driver)

Variables

- struct i2c_client * i2c_client

  Pointer to the i2c_client instance.
- static struct of_device_id driver_id [ ]

  Specifies the ".compatible" strings. of_device_id array should store the same value as corresponding node's "compatible" field in the device tree. In this case the oled.dts in the same directory has the "compatible" field. When the .compatible field here matches the device tree, the I2C device will be probed.
- static struct i2c_device_id driver_device_id [ ]

  This array is pointed by the id_table field of struct i2c_driver. The id_table is used for non-DT based probing of I2C-devices.
- static struct i2c_driver **i2c_driver**

5.3.1 Detailed Description

This file implements the necessary i2c_client probe and remove callbacks on the SSD1306 I2C bus device driver. On top of driver.c, display configurations and initialization are implemented in datalink.c. On top of datalink, OLED printing / graphics are implemented in graphics.c.

5.3.2 Function Documentation

5.3.2.1 driver_on_probe()

```
static int driver_on_probe (
            struct i2c_client * client,
            const struct i2c_device_id * device_id )  [static]
```

Callback function on probing (driver-device binding) of the device driver.

Parameters

client	Pointer to the i2c_client instance.
device_id	The device id to be probed.

Returns

Error status.

5.3.2.2 driver_on_remove()

```
static int driver_on_remove (
            struct i2c_client * client )  [static]
```

Callback function on the removal of the device driver.

Parameters

client	Pointer to the i2c_client instance.

Returns

None.

5.3.3 Variable Documentation

5.3.3.1 driver_device_id

```
struct i2c_device_id driver_device_id[]  [static]
```

Initial value:

```
= {{"oled_device", 0},
                                                    {}}
```

This array is pointed by the id_table field of struct i2c_driver. The id_table is used for non-DT based probing of I2C-devices.

5.3.3.2 driver_id

```
struct of_device_id driver_id[]  [static]
```

Initial value:

```
= {
    {.compatible = "ssd1306, oled_device"}, {}}
```

Specifies the ".compatible" strings. of_device_id array should store the same value as corresponding node's "compatible" field in the device tree. In this case the oled.dts in the same directory has the "compatible" field. When the .compatible field here matches the device tree, the I2C device will be probed.

5.3.3.3 i2c_client

```
struct i2c_client* i2c_client
```

Pointer to the i2c_client instance.

Note

Original symbol declared in driver.c.

5.3.3.4 i2c_driver

```
struct i2c_driver i2c_driver  [static]
```

Initial value:

```
= {

    .probe = driver_on_probe,
    .remove = driver_on_remove,
    .id_table = driver_device_id,
    .driver =
        {
            .name = "oled_device",
            .of_match_table = driver_id,
        },
}
```

5.4 graphics.c File Reference

ssd1306 OLED graphics display APIs implementation.

```
#include "graphics.h"
#include "stdarg.h"
```

Include dependency graph for graphics.c:

Macros

- #define **FONT_TABLE_CHAR_WIDTH** 6
- #define **DEFAULT_MESSAGE_LENGTH** 256
- #define DINOSAUR_BITMAP_ROWS 4

  Bitmap for a dinosaur.
- #define **DINOSAUR_BITMAP_COLUMNS** 32

Functions

- void oled_fill_all (uint8_t pattern)

  Fill the entire screen with byte pattern.
- void oled_set_cursor (uint8_t line, uint8_t position)

  Set the cursor position, i.e. the start location to print.
- void oled_new_line (void)

  Change to a new line on the OLED screen.
- void oled_putc (unsigned char ascii_char)

  Print single char to the oled screen.
- void oled_printf (const char *format,...)

  printf with variadic arguments to print on the oled screen.
- void oled_draw_dino_map (void)

  Draw a dinosaur on the oled screen.

Variables

- static const unsigned char FONT_TABLE [ ][FONT_TABLE_CHAR_WIDTH]

  ASCII Font table defined in hex encoding.
- const unsigned char **DINOSAUR_BITMAP** [DINOSAUR_BITMAP_ROWS][DINOSAUR_BITMAP_COLUMNS]
- static oled_graphics_params_t oled_graphics_params

  Struct that book-keeps parameters for the oled graphics.

5.4.1 Detailed Description

ssd1306 OLED graphics display APIs implementation.

5.4.2 Macro Definition Documentation

5.4.2.1 DINOSAUR_BITMAP_ROWS

```
#define DINOSAUR_BITMAP_ROWS 4
```

Bitmap for a dinosaur.

Note

Bitmap code generated using https://javl.github.io/image2cpp/

5.4.3 Function Documentation

5.4.3.1 oled_draw_dino_map()

```
void oled_draw_dino_map (
            void  )
```

Draw a dinosaur on the oled screen.

Parameters

None.	

Returns

None.

5.4.3.2 oled_fill_all()

```
void oled_fill_all (
            uint8_t pattern )
```

Fill the entire screen with byte pattern.

Parameters

pattern	Byte pattern to fill.

Returns

None.

5.4.3.3 oled_new_line()

```
void oled_new_line (
            void  )
```

Change to a new line on the OLED screen.

Parameters

None.	

Returns

None.

5.4.3.4 oled_printf()

```
void oled_printf (
            const char * format,
             ...  )
```

printf with variadic arguments to print on the oled screen.

Parameters

format	Format supplied including string and/or parameters.

Returns

None.

5.4.3.5 oled_putc()

```
void oled_putc (
            unsigned char ascii_char )
```

Print single char to the oled screen.

Parameters

ascii_char	ASCII character to put.

5.4.3.6 oled_set_cursor()

```
void oled_set_cursor (
            uint8_t line,
            uint8_t position )
```

Set the cursor position, i.e. the start location to print.

Parameters

line	The vertical line (page) to set the cursor to.
position	The horizontal position (column) to the set the cursor to.

5.4.4 Variable Documentation

5.4.4.1 DINOSAUR_BITMAP

```
const unsigned char DINOSAUR_BITMAP[DINOSAUR_BITMAP_ROWS][DINOSAUR_BITMAP_COLUMNS]
```

Initial value:

```
= {
        {0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
         0x00, 0x00, 0x00, 0x00, 0xf0, 0xf8, 0xe8, 0xf8, 0xf8, 0xf8, 0xf8,
         0xf8, 0xf8, 0xf0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00},
        {0x00, 0x00, 0x00, 0x00, 0x00, 0xfc, 0xf0, 0xe0, 0xc0, 0xc0, 0xe0,
         0xf0, 0xf0, 0xf8, 0xfc, 0xff, 0xff, 0xff, 0xff, 0x13, 0x32, 0x02,
         0x02, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00},
```

```
{0x00, 0x00, 0x00, 0x40, 0x40, 0x00, 0x01, 0x03, 0x07, 0x0f, 0xff,
 0xbf, 0x1f, 0x0f, 0x1f, 0xff, 0x87, 0x03, 0x01, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00},
{0xf0, 0xf0, 0xf0, 0xf0, 0xf0, 0xf0, 0xf0, 0xf0, 0xf0, 0xf0, 0xf0,
 0xf0, 0xf0, 0xf0, 0xf0, 0xf0, 0xf0, 0xf0, 0xf0, 0xf0, 0xf0, 0xf0,
 0xf0, 0xf0, 0xf0, 0xf0, 0xf0, 0xf0, 0xf0, 0xf0, 0xf0, 0xf0, 0xf0}}
```

5.4.4.2 FONT_TABLE

```
const unsigned char FONT_TABLE[][FONT_TABLE_CHAR_WIDTH]  [static]
```

ASCII Font table defined in hex encoding.

Note

This table is accessed through numerical value of a char. Each single char is rendered on screen byte by byte (per slice). Non-Alphanumeric characters are encoded 0.

5.4.4.3 oled_graphics_params

```
oled_graphics_params_t oled_graphics_params  [static]
```

Initial value:

```
= {
    .line = 0, .cursor_position = 0, .font_char_width = FONT_TABLE_CHAR_WIDTH}
```

Struct that book-keeps parameters for the oled graphics.

Parameters

line	Current line (page) the cursor is on.
cursor_position	Current position (column) the cursor is on.
font_char_width	ASCII char width for estimation of potential colunm overrun.

5.5 graphics.h File Reference

ssd1306 OLED graphics display APIs header.

```
#include "datalink.h"
```

Include dependency graph for graphics.h: This graph shows which files directly or indirectly include this file:

Classes

- struct oled_graphics_params_t

Macros

- #define **OLED_CANVAS_WIDTH_PIXELS** 128
- #define **OLED_CANVAS_HEIGHT_PIXELS** 64
- #define **BITS_PER_BYTE** 8
- #define **OLED_COLUMN_LENGTH** OLED_CANVAS_WIDTH_PIXELS
- #define **OLED_COLUMN_MIN** 0
- #define **OLED_COLUMN_MAX** OLED_CANVAS_WIDTH_PIXELS - 1
- #define **OLED_PAGE_LENGTH** (OLED_CANVAS_HEIGHT_PIXELS / BITS_PER_BYTE)
- #define **OLED_PAGE_MIN** 0
- #define **OLED_PAGE_MAX** (OLED_CANVAS_HEIGHT_PIXELS / BITS_PER_BYTE) - 1

Functions

- void oled_putc (unsigned char c)

  Print single char to the oled screen.
- void oled_printf (const char ∗format,...)

  printf with variadic arguments to print on the oled screen.
- void oled_new_line (void)

  Change to a new line on the OLED screen.
- void oled_set_cursor (uint8_t lineNo, uint8_t cursorPos)

  Set the cursor position, i.e. the start location to print.
- void oled_fill_all (uint8_t data)

  Fill the entire screen with byte pattern.
- void oled_draw_dino_map (void)

  Draw a dinosaur on the oled screen.

5.5.1 Detailed Description

ssd1306 OLED graphics display APIs header.

5.5.2 Function Documentation

5.5.2.1 oled_draw_dino_map()

```
void oled_draw_dino_map (
            void  )
```

Draw a dinosaur on the oled screen.

Parameters

None.	

Returns

None.

5.5.2.2 oled_fill_all()

```
void oled_fill_all (
            uint8_t pattern )
```

Fill the entire screen with byte pattern.

Parameters

pattern	Byte pattern to fill.

Returns

None.

5.5.2.3 oled_new_line()

```
void oled_new_line (
            void  )
```

Change to a new line on the OLED screen.

Parameters

None.	

Returns

None.

5.5.2.4 oled_printf()

```
void oled_printf (
            const char * format,
             ...  )
```

printf with variadic arguments to print on the oled screen.

Parameters

format	Format supplied including string and/or parameters.

Returns

None.

5.5.2.5 oled_putc()

```
void oled_putc (
            unsigned char ascii_char )
```

Print single char to the oled screen.

Parameters

ascii_char	ASCII character to put.

5.5.2.6 oled_set_cursor()

```
void oled_set_cursor (
            uint8_t line,
            uint8_t position )
```

Set the cursor position, i.e. the start location to print.

Parameters

line	The vertical line (page) to set the cursor to.
position	The horizontal position (column) to the set the cursor to.

Index