SSD1306_OLED_Device_Driver

Generated by Doxygen 1.8.13

Contents

Chapter 1

GNU/Linux Loadable Kernel Module Driver for SSD1306 OLED/PLED Driver.

Source code hierarchy:

```
           oled_sysfs
               |
            graphics
               |
            datalink
               |
         driver  oled.dts
```

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

PDF documents generated (by doxygen) at /docs/latex/refman.pdf To read the compiled pdf, git clone this repository, and open the softlink oled_driver_manual.pdf.

Demo: Displaying text and the dinosaur graphics from chrome browser.

Demo: Displaying oled_sysfs (kobject-mapped directory) from terminal.

Documentation.

To compile.

```
Setup compile environment
$ sudo make setup

Compile
$ sudo make

Successful compile message example:
pi@raspberrypi:~/Projects/raspberrypi-4b/drivers/oled $ make
    make -C /usr/src/linux-headers-5.10.103-v7l+ \
            ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- \
            M=/home/pi/Projects/raspberrypi-4b/drivers/oled modules
    make[1]: Entering directory '/usr/src/linux-headers-5.10.103-v7l+'
      CC [M]  /home/pi/Projects/raspberrypi-4b/drivers/oled/oled_sysfs.o
      LD [M]  /home/pi/Projects/raspberrypi-4b/drivers/oled/oled_driver.o
      MODPOST /home/pi/Projects/raspberrypi-4b/drivers/oled/Module.symvers
      CC [M]  /home/pi/Projects/raspberrypi-4b/drivers/oled/oled_driver.mod.o
      LD [M]  /home/pi/Projects/raspberrypi-4b/drivers/oled/oled_driver.ko
    make[1]: Leaving directory '/usr/src/linux-headers-5.10.103-v7l+'
```

To run:

```
Insert the kernel module
```

```
    $ sudo make insmod
```

To remove the kernel module:

```
    $ sudo make rmmod
```

To check for printk log:

```
    $ 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 function.
  - Constructing Makefile, setup build-environment (linux kernel headers)
  - Understanding struct i2c_client , struct i2c_driver.
  - Implementing probe and remove callbacks when the kernel inserts/remove the driver.
- [x] release-01: Add font / image support to the screen datalink layer.
  - Reading and implementing various display utitlities (change line, clear screen, set coordinate...etc) according to SSD1306 datasheet by Solomon Systech.
- [x] release-02: Add oled_sysfs as an interface to user-space.
  - Understanding struct kobject, kobj_attrbute.
  - Implementing the creation of the oled device as a sysfs folder.
  - Implementing the creation of oled attributes such as display_text, brightness, etc. as files in that sysfs folder.
  - Implementing kthread in driver.c that flushes display_text stored through sysfs onto the screen.
- [ ] release-03: Develop the dinosaur game displaying on this screen.
  - Add multi-threading protection to critical sections.
  - Developing the Chrome dinosaur game, interacting with the kernel module through oled_sysfs.
- [ ] release-04: Unit testing.
  - TBD

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_cursor_coordinate_t Struct Reference

Public Attributes

- uint8_t **line**
- uint8_t **position**

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

- graphics.h

4.2 oled_graphics_params_t Struct Reference

Struct used to book-keep parameters for the oled graphics.

`#include <graphics.h>`

Collaboration diagram for oled_graphics_params_t:

Public Attributes

- oled_cursor_coordinate_t **cursor_coordinate**
- char **display_text** [DEFAULT_TEXT_LENGTH]

4.2.1 Detailed Description

Struct used to book-keep parameters for the oled graphics.

Parameters

cursor_coordinate	Keeps track of the coordinate of current cursor.
display_text	Buffers/keeps track of the current text on the oled_screen.

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

- graphics.h

4.3 Pixel Struct Reference

4.3.1 Detailed Description

the screen.

Parameters

line	The horizontal line (page).
position	The vertical position (column).

Note

Page and columns are formal terms defined in SSD1306 datasheet, Figure 10-3. In this program we have renamed them respectively line and position as more intuitive names. The screen is 128x64 pixels. However, since data are written in slices (one byte) at a time, there are 64 / 8 = 8 lines.

Visualize: ------— 128 positions / columns -------— |||||||| |||||||| ||||..... |||||||| ||||||||

-
-
- 8 lines/pages -
-
-
-
-

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.

Author

Luyao Han (luyaohan1001@gmail.com)

Date

12-21-2022

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.

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.
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.

Note

The I2C bus interface write-data scheme is explained in section 8.1.5.1 in SSD1306 datasheet by Solomon Systech.

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 "oled_sysfs.h"
#include <linux/delay.h>
#include <linux/i2c.h>
#include <linux/kthread.h>
#include <linux/module.h>
#include <linux/sysfs.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 pointer called on probing (driver-device binding) of the device driver. This function implements the following prototype defined struct i2c_driver in linux/i2c.h: int (∗probe)(struct i2c_client ∗client, const struct i2c_device_id ∗id);.
- static int driver_on_remove (struct i2c_client ∗client)

  Callback function pointe called on the removal of the device driver. This function implements the following prototype defined struct i2c_driver in linux/i2c.h: void (∗remove)(struct i2c_client ∗client);.
- static int oled_display_text_thread (void ∗parameters)

  Thread implementing for deploying oled_graphics_params.display_text to oled screen.
- 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

  Identifies the device (i.e. SSD1306 OLED contoller) connected to the i2c bus.
- struct task_struct ∗ handle_display_text_thread

  Points to the oled_display_text_thread created.
- oled_graphics_params_t oled_graphics_params

  Link the symbol to its spawn in graphics.c.
- 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.

Author

Luyao Han (luyaohan1001@gmail.com)

Date

12-21-2022

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 pointer called on probing (driver-device binding) of the device driver. This function implements the following prototype defined struct i2c_driver in linux/i2c.h: int (∗probe)(struct i2c_client ∗client, const struct i2c_device_id ∗id);.

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 pointe called on the removal of the device driver. This function implements the following prototype defined struct i2c_driver in linux/i2c.h: void (∗remove)(struct i2c_client ∗client);.

Parameters

client	Pointer to the i2c_client instance.

Returns

None.

5.3.2.3 oled_display_text_thread()

```
static int oled_display_text_thread (
            void * parameters )  [static]
```

Thread implementing for deploying oled_graphics_params.display_text to oled screen.

Parameters

None.	

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
```

Identifies the device (i.e. SSD1306 OLED contoller) connected to the i2c bus.

Pointer to the i2c_client instance.

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,
        },
}
```

5.3.3.5 oled_graphics_params

```
oled_graphics_params_t oled_graphics_params
```

Link the symbol to its spawn in graphics.c.

Link the symbol to its spawn in graphics.c.

Parameters

cursor_coordinate	Keeps track of the coordinate of current cursor.
display_text	Buffers/keeps track of the current text on the oled_screen.

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_CHAR_WIDTH** 6
- #define **ASCII_TABLE_LENGTH** 128
- #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 (oled_cursor_coordinate_t cursor_coordinate)

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

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

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

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

  Draw a dinosaur on the oled screen.

Variables

- static const unsigned char FONT_TABLE [ASCII_TABLE_LENGTH][FONT_CHAR_WIDTH]

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

  Book-keeps parameters for the oled graphics.

5.4.1 Detailed Description

ssd1306 OLED graphics display APIs implementation.

Author

Luyao Han (luyaohan1001@gmail.com)

Date

12-21-2022

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 (
            oled_cursor_coordinate_t cursor_coordinate )
```

Draw a dinosaur on the oled screen.

Parameters

cursor_coordinate	Set to this coordinate as the start pixel and draw the dinosaur.

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 (
            oled_new_line_options new_line_option )
```

Change to a new line on the OLED screen.

Parameters

oled_new_line_options	START_OF_NEW_LINE to print to the start of the new line. SAME_CURSOR_POSITION to print the next line the same cursor position.

Returns

None.

5.4.3.4 oled_printf()

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

printf on oled 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 )
```

Put single char to the oled screen.

Print single char to the oled screen.

Parameters

ascii_char	ASCII character to put.

Returns

None.

5.4.3.6 oled_set_cursor()

```
void oled_set_cursor (
            oled_cursor_coordinate_t cursor_coordinate )
```

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

Parameters

cursor_coordinate	The pixel coordinate to 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}}
```

5.4.4.2 FONT_TABLE

```
const unsigned char FONT_TABLE[ASCII_TABLE_LENGTH][FONT_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; they are meaningless for printing but including them avoids remapping when interpreting ascii numeric value as the access index to this table.

5.4.4.3 oled_graphics_params

```
oled_graphics_params_t oled_graphics_params
```

Initial value:

```
= {
    .cursor_coordinate = {.line = 0, .position = 0}, .display_text = "\0"}
```

Book-keeps parameters for the oled graphics.

Link the symbol to its spawn in graphics.c.

Parameters

cursor_coordinate	Keeps track of the coordinate of current cursor.
display_text	Buffers/keeps track of the current text on the oled_screen.

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_cursor_coordinate_t
- struct oled_graphics_params_t

  Struct used to book-keep parameters for the oled graphics.

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
- #define **DEFAULT_TEXT_LENGTH** 256

Enumerations

- enum oled_new_line_options { **START_OF_NEW_LINE**, **SAME_CURSOR_POSITION** }

  Enum type defining options when printing to a new line (page).

Functions

- void oled_putc (unsigned char c)

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

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

  Change to a new line on the OLED screen.
- void oled_set_cursor (oled_cursor_coordinate_t cursor_coordinate)

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

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

  Draw a dinosaur on the oled screen.

5.5.1 Detailed Description

SSD1306 OLED graphics display APIs header.

Author

Luyao Han (luyaohan1001@gmail.com)

Date

12-21-2022

5.5.2 Enumeration Type Documentation

5.5.2.1 oled_new_line_options

```
enum oled_new_line_options
```

Enum type defining options when printing to a new line (page).

Parameters

START_OF_NEW_LINE	For example, when printing a sentence, it is expected to be printed to the start of the new line.
SAME_CURSOR_POSITION	For exapmle, when printing an image, it is expected to be printed to the same cursor position.

5.5.3 Function Documentation

5.5.3.1 oled_draw_dino_map()

```
void oled_draw_dino_map (
            oled_cursor_coordinate_t cursor_coordinate )
```

Draw a dinosaur on the oled screen.

Parameters

cursor_coordinate	Set to this coordinate as the start pixel drawing the dinosaur.

Returns

None.

Parameters

cursor_coordinate	Set to this coordinate as the start pixel and draw the dinosaur.

Returns

None.

5.5.3.2 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.

5.5.3.3 oled_new_line()

```
void oled_new_line (
            oled_new_line_options new_line_option )
```

Change to a new line on the OLED screen.

Parameters

oled_new_line_options	START_OF_NEW_LINE to print to the start of the new line. SAME_CURSOR_POSITION to print the next line the same cursor position.

Returns

None.

5.5.3.4 oled_printf()

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

printf on oled with variadic arguments to print on the oled screen.

Parameters

format	Format supplied including string and/or parameters.

Returns

None.

5.5.3.5 oled_putc()

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

Print single char to the oled screen.

Parameters

ascii_char	ASCII character to put.

Returns

None.

Print single char to the oled screen.

Parameters

ascii_char	ASCII character to put.

Returns

None.

5.5.3.6 oled_set_cursor()

```
void oled_set_cursor (
            oled_cursor_coordinate_t cursor_coordinate )
```

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

Parameters

cursor_coordinate	The pixel coordinate to set the cursor to.

5.6 oled_sysfs.c File Reference

Init/deinit callbacks implementation to expose user-control through sysfs filesystem.

```
#include "oled_sysfs.h"
#include "graphics.h"
#include <linux/kobject.h>
```

Include dependency graph for oled_sysfs.c:

Functions

- static ssize_t kobj_attr_display_text_show (struct kobject ∗kobj, struct kobj_attribute ∗attr, char ∗buffer)

  Callback function prototype for when the user read display_text, i.e. cat /sys/kernel/oled_sysfs/display_text. The prototype implements the following function pointer in struct kobj_attribute in linux/kobject.h: ssize_t (∗show)(struct kobject ∗kobj, struct kobj_attribute ∗attr, char ∗buf);.
- static ssize_t kobj_attr_display_text_store (struct kobject ∗kobj, struct kobj_attribute ∗attr, const char ∗buffer, size_t count)

  Callback function prototype for when the user write to the display_text, i.e. echo "hello, world" > /sys/kernel/oled_↩ sysfs/display_text. The prototype implements the following function pointer in struct kobj_attribute in linux/object.h.
- int oled_sysfs_init (void)

  Creates kobject and its attributes under sysfs.
- void oled_sysfs_deinit (void)

  Cleans up the constructs created in oled_sysfs_init. Deletes the kernel object allocated and the sysfs folder created for oled_kobj.

Variables

- struct kobject ∗ oled_kobj

  The pointer storing a oled kernel object to be created later.
- oled_graphics_params_t oled_graphics_params

  Link the symbol to its spawn in graphics.c.
- static struct kobj_attribute kobj_attr_display_text

  "display_text" attribute, storing the current text the oled displaying.

5.6.1 Detailed Description

Init/deinit callbacks implementation to expose user-control through sysfs filesystem.

Headers to expose user-control through sysfs filesystem.

Author

Luyao Han (luyaohan1001@gmail.com)

Date

12-21-2022

5.6.2 Function Documentation

5.6.2.1 kobj_attr_display_text_show()

```
static ssize_t kobj_attr_display_text_show (
            struct kobject * kobj,
            struct kobj_attribute * attr,
            char * buffer )  [static]
```

Callback function prototype for when the user read display_text, i.e. cat /sys/kernel/oled_sysfs/display_text. The prototype implements the following function pointer in struct kobj_attribute in linux/kobject.h: ssize_t (∗show)(struct kobject ∗kobj, struct kobj_attribute ∗attr, char ∗buf);.

Parameters

kobj	Kobject to which tied sysfs file is read (show).
attr	Attr.com/ibute to which the tied sysfs file is read (show).
buffer	Text display to the screen when the file is read.

Returns

Error status.

5.6.2.2 kobj_attr_display_text_store()

```
static ssize_t kobj_attr_display_text_store (
            struct kobject * kobj,
            struct kobj_attribute * attr,
            const char * buffer,
            size_t count )  [static]
```

Callback function prototype for when the user write to the display_text, i.e. echo "hello, world" > /sys/kernel/oled↩ _sysfs/display_text. The prototype implements the following function pointer in struct kobj_attribute in linux/object.h.

Parameters

kobj	Kobject to which tied sysfs file is written (store).
attr	Attribute to which the tied sysfs file is written (store).
buffer	Text display to the screen when the file is written.

Returns

Number of characters written.

Note

Returning status code is wrong, and could cause the system looping in store function.

5.6.2.3 oled_sysfs_deinit()

```
void oled_sysfs_deinit (
            void  )
```

Cleans up the constructs created in oled_sysfs_init. Deletes the kernel object allocated and the sysfs folder created for oled_kobj.

Parameters

None.	

Returns

None.

5.6.2.4 oled_sysfs_init()

```
int oled_sysfs_init (
            void  )
```

Creates kobject and its attributes under sysfs.

Parameters

None.	

Returns

status_code.

5.6.3 Variable Documentation

5.6.3.1 kobj_attr_display_text

```
struct kobj_attribute kobj_attr_display_text  [static]
```

Initial value:

```
= {
    .attr = {.name = "display_text", .mode = 0666},
    .show = kobj_attr_display_text_show,
    .store = kobj_attr_display_text_store}
```

"display_text" attribute, storing the current text the oled displaying.

Note

"display_text" will show up as a file under /sys/kernel/oled_sysfs.

5.6.3.2 oled_graphics_params

```
oled_graphics_params_t oled_graphics_params
```

Link the symbol to its spawn in graphics.c.

Link the symbol to its spawn in graphics.c.

Parameters

cursor_coordinate	Keeps track of the coordinate of current cursor.
display_text	Buffers/keeps track of the current text on the oled_screen.

5.6.3.3 oled_kobj

```
struct kobject* oled_kobj
```

The pointer storing a oled kernel object to be created later.

Note

The kobject, once created, will show up as a directory under /sys/kernel/.

Index