

Main Page	Data Structures	Files	Examples
Data Structures	Data Fields		

# Data Structures

Here are the data structures with brief descriptions:

<b>fiosd_char</b>	This structure defines for add or replace a character in OSD ram.  The index 0 is special for FIOSDS_RMCHAR
<b>fiosd_charmap</b>	This structure defines for getting the bit map of characters
<b>fiosd_font_info</b>	This structure defines for setting OSD font feature.  If width and height smaller than fonts in OSD ram, the display font will be cut. For example: If the fonts size is 12x16 in sram, we set the <b>fiosd_font_info</b> as 6x8. The OSD will display the quarter font from top-left
<b>fiosd_hw_info</b>	This structure defines for getting OSD hardware information
<b>fiosd_palette</b>	This structure defines for setting palette color
<b>fiosd_string</b>	This structure defines for setting display string.  The OSD will display string from fiosd_string->start to fiosd_string->start + strlen(fiosd_string->string) in OSD display array. The string[128] is string index that depend on char bit map. While using multiple OSD windows at the same time, user has to manager string length in the common string buffer
<b>fiosd_transparent</b>	This structure defines for setting transparent

	level
<b>fiosd_win</b>	This structure defines for setting OSD window position and size
<b>fiosdmask_win</b>	This structure defines for setting mask window position and size

---

Generated on Thu Nov 25 2010 17:50:15 for GM8126/GM812X Capture IOCTL and API Reference by [doxygen](#) 1.7.1

Main Page	Data Structures	Files	Examples
Data Structures	Data Fields	Data Fields	

## **fiosd\_char Struct Reference**

---

This structure defines for add or replace a character in OSD ram.

The index 0 is special for FIOSDS\_RMCHAR. [More...](#)

## Data Fields

---

char	<b>font</b>
uint8_t	<b>fbitmap</b> [36]

---

## Detailed Description

This structure defines for add or replace a character in OSD ram.

The index 0 is special for FIOSDS\_RMCHAR.

### Examples:

[cap\\_osd\\_string.c](#).

Definition at line [84](#) of file [vcap\\_osd.h](#).

---

## Field Documentation

### **char font**

the font index that you want to setting or replacing

#### **Examples:**

[cap\\_osd\\_string.c](#).

Definition at line **85** of file [vcap\\_osd.h](#).

### **uint8\_t fbitmap[36]**

the font bitmap

#### **Examples:**

[cap\\_osd\\_string.c](#).

Definition at line **86** of file [vcap\\_osd.h](#).

Main Page	Data Structures	Files	Examples
Data Structures	Data Fields	Data Fields	

## **fiosd\_charmap Struct Reference**

---

This structure defines for getting the bit map of characters. [More...](#)

## Data Fields

uint32\_t **map** [8]

---

## Detailed Description

This structure defines for getting the bit map of characters.

Definition at line [120](#) of file [vcap\\_osd.h](#).

---

## Field Documentation

**uint32\_t map[8]**

the bitmap of fonts that store in OSD ram

Definition at line [121](#) of file [vcap\\_osd.h](#).

---

Generated on Thu Nov 25 2010 17:50:15 for GM8126/GM812X Capture IOCTL and API

Reference by [doxygen](#) 1.7.1

Main Page	Data Structures	Files	Examples
Data Structures	Data Fields	Data Fields	

## **fiosd\_font\_info Struct Reference**

---

This structure defines for setting OSD font feature.

If width and height smaller than fonts in OSD ram, the display font will be cut.

For example:

If the fonts size is 12x16 in sram, we set the **fiosd\_font\_info** as 6x8.

The OSD will display the quarter font from top-left. [More...](#)

## Data Fields

---

uint32_t	<b>width</b> :8
uint32_t	<b>height</b> :8
uint32_t	<b>row_space</b> :8
uint32_t	<b>col_space</b> :8
uint8_t	<b>windex</b>

---

## Detailed Description

This structure defines for setting OSD font feature.

If width and height smaller than fonts in OSD ram, the display font will be cut.

For example:

If the fonts size is 12x16 in sram, we set the **fiosd\_font\_info** as 6x8.  
The OSD will display the quarter font from top-left.

Definition at line **61** of file **vcap\_osd.h**.

---

## Field Documentation

### `uint32_t width`

the font width that will be display

Definition at line [62](#) of file [vcap\\_osd.h](#).

### `uint32_t height`

the font height that will be display

Definition at line [63](#) of file [vcap\\_osd.h](#).

### `uint32_t row_space`

the font row space

Definition at line [64](#) of file [vcap\\_osd.h](#).

### `uint32_t col_space`

the font column space

Definition at line [65](#) of file [vcap\\_osd.h](#).

### `uint8_t windex`

the OSD window index

Definition at line [66](#) of file [vcap\\_osd.h](#).

Generated on Thu Nov 25 2010 17:50:15 for GM8126/GM812X Capture IOCTL and API  
Reference by [doxygen](#) 1.7.1

Main Page	Data Structures	Files	Examples
Data Structures	Data Fields	Data Fields	

## **fiosd\_hw\_info Struct Reference**

---

This structure defines for getting OSD hardware information. More...

## Data Fields

int **MaxFontNum**

int **MaxDispNum**

---

## Detailed Description

This structure defines for getting OSD hardware information.

Definition at line [139](#) of file [vcap\\_osd.h](#).

---

## Field Documentation

### **int MaxFontNum**

max OSD font number

Definition at line [140](#) of file [vcap\\_osd.h](#).

### **int MaxDispNum**

max OSD display number

Definition at line [141](#) of file [vcap\\_osd.h](#).

---

Generated on Thu Nov 25 2010 17:50:15 for GM8126/GM812X Capture IOCTL and API

Reference by [doxygen](#) 1.7.1

Main Page	Data Structures	Files	Examples
Data Structures	Data Fields	Data Fields	

## **fiosd\_palette Struct Reference**

---

This structure defines for setting palette color. [More...](#)

## Data Fields

---

uint8_t	<b>index</b>
uint8_t	<b>y</b>
uint8_t	<b>cb</b>
uint8_t	<b>cr</b>

---

## Detailed Description

This structure defines for setting palette color.

### Examples:

[cap\\_osd\\_mask.c](#), and [cap\\_osd\\_string.c](#).

Definition at line [72](#) of file [vcap\\_osd.h](#).

---

## Field Documentation

### `uint8_t index`

the palette color index. The value is from 0 to 6

#### **Examples:**

[cap\\_osd\\_mask.c](#), and [cap\\_osd\\_string.c](#).

Definition at line [73](#) of file [vcap\\_osd.h](#).

### `uint8_t y`

the color of Y

#### **Examples:**

[cap\\_osd\\_mask.c](#), and [cap\\_osd\\_string.c](#).

Definition at line [74](#) of file [vcap\\_osd.h](#).

### `uint8_t cb`

the color of Cb

#### **Examples:**

[cap\\_osd\\_mask.c](#), and [cap\\_osd\\_string.c](#).

Definition at line [75](#) of file [vcap\\_osd.h](#).

### `uint8_t cr`

the color of Cr

#### **Examples:**

`cap_osd_mask.c`, and `cap_osd_string.c`.

Definition at line **76** of file `vcap_osd.h`.

---

Generated on Thu Nov 25 2010 17:50:15 for GM8126/GM812X Capture IOCTL and API

Reference by [doxygen](#) 1.7.1

Main Page	Data Structures	Files	Examples
Data Structures	Data Fields	Data Fields	

## **fiosd\_string Struct Reference**

---

This structure defines for setting display string.

The OSD will display string from fiosd\_string->start to fiosd\_string->start +  
strlen(fiosd\_string->string) in OSD display array. The string[128] is  
string index that depend on char bit map. While using multiple OSD windows  
at the same time, user has to manager string length in the common string buffer.  
[More...](#)

## Data Fields

---

uint8_t	windex
uint8_t	start
uint8_t	fg_color
uint8_t	bg_color
char	string [128]

---

## Detailed Description

This structure defines for setting display string.

The OSD will display string from fiosd\_string->start to fiosd\_string->start + strlen(fiosd\_string->string) in OSD display array. The string[128] is string index that depend on char bit map. While using multiple OSD windows at the same time, user has to manager string length in the common string buffer.

### Examples:

[cap\\_osd\\_string.c](#).

Definition at line [97](#) of file [vcap\\_osd.h](#).

---

## Field Documentation

### **uint8\_t windex**

the OSD window index

Definition at line **98** of file [vcap\\_osd.h](#).

### **uint8\_t start**

the string start index in OSD display array. 0-255

Definition at line **99** of file [vcap\\_osd.h](#).

### **uint8\_t fg\_color**

the forward color index that assign by FIOSDS\_PALTCOLOR

Definition at line **100** of file [vcap\\_osd.h](#).

### **uint8\_t bg\_color**

the backward color index that assign by FIOSDS\_PALTCOLOR

Definition at line **101** of file [vcap\\_osd.h](#).

### **char string[128]**

the string that will be displayed

Definition at line **102** of file [vcap\\_osd.h](#).

Generated on Thu Nov 25 2010 17:50:15 for GM8126/GM812X Capture IOCTL and API  
Reference by [doxygen](#) 1.7.1

Main Page	Data Structures	Files	Examples
Data Structures	Data Fields	Data Fields	

## **fiosd\_transparent Struct Reference**

---

This structure defines for setting transparent level. [More...](#)

## Data Fields

`uint8_t windex`

`uint8_t level`

---

## **Detailed Description**

This structure defines for setting transparent level.

### **Examples:**

[\*\*cap\\_osd\\_mask.c\*\*](#), and [\*\*cap\\_osd\\_string.c\*\*](#).

Definition at line [\*\*108\*\*](#) of file [\*\*vcap\\_osd.h\*\*](#).

---

## Field Documentation

### `uint8_t windex`

the OSD window index

**Examples:**

[cap\\_osd\\_mask.c](#), and [cap\\_osd\\_string.c](#).

Definition at line [109](#) of file [vcap\\_osd.h](#).

### `uint8_t level`

the transparent level 0, 50, 75, 100 percent

**Examples:**

[cap\\_osd\\_mask.c](#), and [cap\\_osd\\_string.c](#).

Definition at line [110](#) of file [vcap\\_osd.h](#).

Main Page	Data Structures	Files	Examples
Data Structures	Data Fields	Data Fields	

## **fiosd\_win Struct Reference**

---

This structure defines for setting OSD window position and size. [More...](#)

## Data Fields

---

uint32\_t **x**:16  
uint32\_t **y**:16  
uint32\_t **hdim**:16  
uint32\_t **vdim**:16  
  uint8\_t **windex**

---

## Detailed Description

This structure defines for setting OSD window position and size.

### Examples:

[cap\\_osd\\_string.c](#).

Definition at line [45](#) of file [vcap\\_osd.h](#).

---

## Field Documentation

### `uint32_t x`

left position of OSD window

**Examples:**

[cap\\_osd\\_string.c](#).

Definition at line **46** of file [vcap\\_osd.h](#).

### `uint32_t y`

top position of OSD window

**Examples:**

[cap\\_osd\\_string.c](#).

Definition at line **47** of file [vcap\\_osd.h](#).

### `uint32_t hdim`

the dimension width of OSD window

**Examples:**

[cap\\_osd\\_string.c](#).

Definition at line **48** of file [vcap\\_osd.h](#).

### `uint32_t vdim`

the dimension height of OSD window

**Examples:**

[\*\*cap\\_osd\\_string.c\*\*](#)

Definition at line **49** of file [\*\*vcap\\_osd.h\*\*](#).

**uint8\_t windex**

the OSD window index

**Examples:**

[\*\*cap\\_osd\\_string.c\*\*](#)

Definition at line **50** of file [\*\*vcap\\_osd.h\*\*](#).

---

Generated on Thu Nov 25 2010 17:50:15 for GM8126/GM812X Capture IOCTL and API

Reference by [\*\*doxygen\*\*](#) 1.7.1

Main Page	Data Structures	Files	Examples
Data Structures	Data Fields	Data Fields	

## **fiosdmask\_win Struct Reference**

---

This structure defines for setting mask window position and size.

[More...](#)

## Data Fields

---

uint32_t	x:16
uint32_t	y:16
uint32_t	width:16
uint32_t	height:16
uint8_t	windex
uint8_t	color

---

## Detailed Description

This structure defines for setting mask window position and size.

### Examples:

[cap\\_osd\\_mask.c](#)

Definition at line [127](#) of file [vcap\\_osd.h](#).

---

## Field Documentation

### `uint32_t x`

left position of mask window

**Examples:**

[cap\\_osd\\_mask.c](#).

Definition at line [128](#) of file [vcap\\_osd.h](#).

### `uint32_t y`

top position of mask window

**Examples:**

[cap\\_osd\\_mask.c](#).

Definition at line [129](#) of file [vcap\\_osd.h](#).

### `uint32_t width`

the dimension width of mask window

**Examples:**

[cap\\_osd\\_mask.c](#).

Definition at line [130](#) of file [vcap\\_osd.h](#).

### `uint32_t height`

the dimension height of mask window

**Examples:**

[\*\*cap\\_osd\\_mask.c\*\*](#)

Definition at line [131](#) of file [\*\*vcap\\_osd.h\*\*](#).

### **uint8\_t windex**

the mask window index

**Examples:**

[\*\*cap\\_osd\\_mask.c\*\*](#)

Definition at line [132](#) of file [\*\*vcap\\_osd.h\*\*](#).

### **uint8\_t color**

the mask color index that assign by FIOSDS\_PALTCOLOR

**Examples:**

[\*\*cap\\_osd\\_mask.c\*\*](#)

Definition at line [133](#) of file [\*\*vcap\\_osd.h\*\*](#).

Main Page	Data Structures	Files	Examples									
Data Structures	Data Fields											
All	Variables											
b	c	f	h	i	I	m	r	s	v	w	x	y

Here is a list of all struct and union fields with links to the structures/unions they belong to:

#### - b -

- bg\_color : [fiosd\\_string](#)

#### - c -

- cb : [fiosd\\_palette](#)
- col\_space : [fiosd\\_font\\_info](#)
- color : [fiosdmask\\_win](#)
- cr : [fiosd\\_palette](#)

#### - f -

- fbitmap : [fiosd\\_char](#)
- fg\_color : [fiosd\\_string](#)
- font : [fiosd\\_char](#)

#### - h -

- hdim : [fiosd\\_win](#)
- height : [fiosdmask\\_win](#) , [fiosd\\_font\\_info](#)

#### - i -

- index : [fiosd\\_palette](#)

#### - I -

- level : [fiosd\\_transparent](#)

#### - m -

- map : **fiosd\_charmap**
- MaxDispNum : **fiosd\_hw\_info**
- MaxFontNum : **fiosd\_hw\_info**

- **r** -

- row\_space : **fiosd\_font\_info**

- **s** -

- start : **fiosd\_string**
- string : **fiosd\_string**

- **v** -

- vdim : **fiosd\_win**

- **w** -

- width : **fiosd\_font\_info** , **fiosdmask\_win**
- windex : **fiosd\_transparent** , **fiosd\_string** , **fiosd\_font\_info** ,  
**fiosd\_win** , **fiosdmask\_win**

- **x** -

- x : **fiosd\_win** , **fiosdmask\_win**

- **y** -

- y : **fiosd\_win** , **fiosdmask\_win** , **fiosd\_palette**

Main Page

Data Structures

Files

Examples

File List

Globals

# File List

---

Here is a list of all files with brief descriptions:

**source/vcap\_osd.h**  
[code]

This file provides the capture OSD and Mask  
related function calls or IOCTLs

---

Generated on Thu Nov 25 2010 17:50:15 for GM8126/GM812X Capture IOCTL and API

Reference by [doxygen](#) 1.7.1

Main Page

Data Structures

Files

Examples

File List

Globals

Data Structures | Defines |  
Typedefs

## **source/vcap\_osd.h File Reference**

---

This file provides the capture OSD and Mask related function calls or IOCTLs. [More...](#)

[Go to the source code of this file.](#)

## Data Structures

---

### struct **fiosd\_win**

This structure defines for setting OSD window position and size. [More...](#)

---

### struct **fiosd\_font\_info**

This structure defines for setting OSD font feature.

If width and height smaller than fonts in OSD ram, the display font will be cut.

For example:

If the font size is 12x16 in sram, we set the **fiosd\_font\_info** as 6x8.

The OSD will display the quarter font from top-left. [More...](#)

---

### struct **fiosd\_palette**

This structure defines for setting palette color. [More...](#)

---

### struct **fiosd\_char**

This structure defines for add or replace a character in OSD ram.

The index 0 is special for FIOSDS\_RMCHAR. [More...](#)

---

### struct **fiosd\_string**

This structure defines for setting display string.

The OSD will display string from fiosd\_string->start to fiosd\_string->start +

strlen(fiosd\_string->string) in OSD display array. The string[128] is string

index that depend on char bit map. While using multiple OSD windows at the same

time, user has to manage string length in the common string buffer. [More...](#)

---

### struct **fiosd\_transparent**

This structure defines for setting transparent level. [More...](#)

---

### struct **fiosd\_charmap**

This structure defines for getting the bit map of characters.

[More...](#)

---

struct **fiosdmask\_win**

This structure defines for setting mask window position and size. [More...](#)

---

struct **fiosd\_hw\_info**

This structure defines for getting OSD hardware information. [More...](#)

## Defines

```
#define MAX_TARGET 10000
#define X2OFFSET(base_w, x) (MAX_TARGET * x / base_w)
#define Y2OFFSET(base_h, y) (MAX_TARGET * y / base_h)
#define W2OFFSET(base_w, w) (MAX_TARGET * w / base_w)
#define H2OFFSET(base_h, h) (MAX_TARGET * h / base_h)
#define OFFSET2X(base_w, off_x) (off_x * base_w / MAX_TARGET)
#define OFFSET2Y(base_h, off_y) (off_y * base_h / MAX_TARGET)
#define OFFSET2W(base_w, off_w) (off_w * base_w / MAX_TARGET)
#define OFFSET2H(base_h, off_h) (off_h * base_h / MAX_TARGET)
#define FOSD_TRANSPARENT_0PERCENT 0
#define FOSD_TRANSPARENT_50PERCENT 1
#define FOSD_TRANSPARENT_75PERCENT 2
#define FOSD_TRANSPARENT_100PERCENT 3
#define VCAP_IOC_MAGIC 'f'
#define FIOSDS_ON _IOW(VCAP_IOC_MAGIC, 1, int)
#define FIOSDS_OFF _IOW(VCAP_IOC_MAGIC, 2, int)
#define FIOSDS_WIN _IOW(VCAP_IOC_MAGIC, 3, fiosd_win_t)
#define FIOSDS_FONTSETTING _IOW(VCAP_IOC_MAGIC, 4,
fiosd_font_info_t)
#define FIOSDS_TRANSPARENT _IOW(VCAP_IOC_MAGIC, 5,
fiosd_transparent_t)
#define FIOSDS_CHAR _IOW(VCAP_IOC_MAGIC, 6,
fiosd_char_t)
#define FIOSDS_STRING _IOW(VCAP_IOC_MAGIC, 7,
fiosd_string_t)
#define FIOSDS_PALTCOLOR _IOW(VCAP_IOC_MAGIC, 8,
fiosd_palette_t)
#define FIOSDS_RMCHAR _IOW(VCAP_IOC_MAGIC, 9, char)
#define FIOSDG_CHARMAP _IOR(VCAP_IOC_MAGIC, 10,
fiosd_charmap_t)
#define FIOSDG_HWINFO _IOR(VCAP_IOC_MAGIC, 11,
fiosd_hw_info_t)
```

```
#define FIOSDS_BWINOFFSET _IOR(VCAP_IOC_MAGIC, 12, int)
#define FIOSDS_FRAMEMODE _IOR(VCAP_IOC_MAGIC, 13, int)
#define FIOSDMASKS_ON _IOW(VCAP_IOC_MAGIC, 30, int)
#define FIOSDMASKS_OFF _IOW(VCAP_IOC_MAGIC, 31, int)
#define FIOSDMASKS_TRANSPARENT _IOW(VCAP_IOC_MAGIC,
32, fiosd_transparent_t)
#define FIOSDMASKS_WIN _IOW(VCAP_IOC_MAGIC, 33,
fiosmask_win_t)
#define FIOSDMASKS_BWINOFFSET _IOW(VCAP_IOC_MAGIC,
34, int)
```

## TypeDefs

---

```
typedef struct fiosd_win fiosd_win_t
typedef struct fiosd_font_info fiosd_font_info_t
typedef struct fiosd_palette fiosd_palette_t
typedef struct fiosd_char fiosd_char_t
typedef struct fiosd_string fiosd_string_t
typedef struct fiosd_transparent fiosd_transparent_t
typedef struct fiosd_charmap fiosd_charmap_t
typedef struct fiosdmask_win fiosdmask_win_t
typedef struct fiosd_hw_info fiosd_hw_info_t
```

---

## Detailed Description

This file provides the capture OSD and Mask related function calls or IOCTLs.

**Author:**

Francis Huang

**Version:**

0.1.0

**Date:**

2010/09/28

Definition in file [vcap\\_osd.h](#).

---

## Define Documentation

```
#define MAX_TARGET 10000
```

Definition at line 30 of file [vcap\\_osd.h](#).

```
#define X2OFFSET( base_w,  
                x  
            ) (MAX_TARGET * x / base_w)
```

Definition at line 32 of file [vcap\\_osd.h](#).

```
#define Y2OFFSET( base_h,  
                y  
            ) (MAX_TARGET * y / base_h)
```

Definition at line 33 of file [vcap\\_osd.h](#).

```
#define W2OFFSET( base_w,  
                w  
            ) (MAX_TARGET * w / base_w)
```

Definition at line 34 of file [vcap\\_osd.h](#).

```
#define H2OFFSET( base_h,  
                h  
            ) (MAX_TARGET * h / base_h)
```

Definition at line 35 of file [vcap\\_osd.h](#).

```
#define OFFSET2X( base_w,  
                 off_x  
             )           (off_x * base_w / MAX_TARGET)
```

Definition at line 37 of file [vcap\\_osd.h](#).

```
#define OFFSET2Y( base_h,  
                 off_y  
             )           (off_y * base_h / MAX_TARGET)
```

Definition at line 38 of file [vcap\\_osd.h](#).

```
#define OFFSET2W( base_w,  
                 off_w  
             )           (off_w * base_w / MAX_TARGET)
```

Definition at line 39 of file [vcap\\_osd.h](#).

```
#define OFFSET2H( base_h,  
                 off_h  
             )           (off_h * base_h / MAX_TARGET)
```

Definition at line 40 of file [vcap\\_osd.h](#).

```
#define FOSD_TRANSPARENT_0PERCENT 0
```

Definition at line 111 of file [vcap\\_osd.h](#).

```
#define FOSD_TRANSPARENT_50PERCENT 1
```

Definition at line 112 of file [vcap\\_osd.h](#).

```
#define FOSD_TRANSPARENT_75PERCENT 2
```

Definition at line [113](#) of file `vcap_osd.h`.

```
#define FOSD_TRANSPARENT_100PERCENT 3
```

Definition at line [114](#) of file `vcap_osd.h`.

```
#define VCAP_IOC_MAGIC 'f'
```

Definition at line [145](#) of file `vcap_osd.h`.

```
#define FIOSDS_ON _IOW(VCAP_IOC_MAGIC, 1, int)
```

Use to enable OSD window

```
ioctl(osd_fd, FIOSDS_ON, &windex).
```

- parameter :

`osd_fd` : OSD device handler

`windex` : specify to enable which OSD window, from 0~3

**Examples:**

`cap_osd_string.c`.

Definition at line [159](#) of file `vcap_osd.h`.

```
#define FIOSDS_OFF _IOW(VCAP_IOC_MAGIC, 2, int)
```

Use to disable OSD window

```
ioctl(osd_fd, FIOSDS_OFF, &windex).
```

- parameter :  
*osd\_fd* : OSD device handler  
*windex* : specify to disable which OSD window, from 0~3

**Examples:**

[cap\\_osd\\_string.c](#).

Definition at line [169](#) of file [vcap\\_osd.h](#).

---

```
#define FIOSDS_WIN _IOW(VCAP_IOC_MAGIC, 3, fiosd_win_t)
```

Use to setup OSD window size and position

`ioctl(osd_fd, FIOSDS_WIN, &win).`

- parameter :  
*osd\_fd* : OSD device handler  
*win* : reference to fiosd\_win\_t structure

**Examples:**

[cap\\_osd\\_string.c](#).

Definition at line [179](#) of file [vcap\\_osd.h](#).

---

```
#define FIOSDS_FONTSETTING _IOW(VCAP_IOC_MAGIC, 4, fiosd_font_info_t)
```

Use to setup OSD font feature

`ioctl(osd_fd, FIOSDS_FONTSETTING, &font_info).`

- parameter :  
*osd\_fd* : OSD device handler  
*font\_info* : reference to fiosd\_font\_info\_t structure

Definition at line [189](#) of file [vcap\\_osd.h](#).

```
#define FIOSDS_TRANSPARENT _IOW(VCAP_IOC_MAGIC, 5, fios
```

Use to setup OSD font transparent level

```
ioctl(osd_fd, FIOSDS_TRANSPARENT, &font_tran).
```

- parameter :

*osd\_fd* : OSD device handler

*font\_tran* : reference to fiosd\_transparent\_t structure

**Examples:**

[cap\\_osd\\_string.c](#).

Definition at line [199](#) of file [vcap\\_osd.h](#).

```
#define FIOSDS_CHAR _IOW(VCAP_IOC_MAGIC, 6, fiosd_char_t)
```

Use to add or replace a character in OSD ram

```
ioctl(osd_fd, FIOSDS_CHAR, &font_char).
```

- parameter :

*osd\_fd* : OSD device handler

*font\_char* : reference to fiosd\_char\_t structure

**Examples:**

[cap\\_osd\\_string.c](#).

Definition at line [209](#) of file [vcap\\_osd.h](#).

```
#define FIOSDS_STRING _IOW(VCAP_IOC_MAGIC, 7, fiosd_string)
```

Use to setup OSD display string

```
ioctl(osd_fd, FIOSDS_STRING, &font_string).
```

- parameter :

***osd\_fd*** : OSD device handler  
***font\_string*** : reference to fiosd\_string\_t structure

**Examples:**

[cap\\_osd\\_string.c](#).

Definition at line [219](#) of file [vcap\\_osd.h](#).

**#define FIOSDS\_PALTCOLOR \_IOW(VCAP\_IOC\_MAGIC, 8, fiosd\_**

Use to setup OSD palette color

**ioctl(osd\_fd, FIOSDS\_PALTCOLOR, &palette).**

- parameter :

***osd\_fd*** : OSD device handler

***palette*** : reference to fiosd\_palette\_t structure

**Examples:**

[cap\\_osd\\_mask.c](#), and [cap\\_osd\\_string.c](#).

Definition at line [229](#) of file [vcap\\_osd.h](#).

**#define FIOSDS\_RMCHAR \_IOW(VCAP\_IOC\_MAGIC, 9, char)**

Use to remove a character from OSD ram

**ioctl(osd\_fd, FIOSDS\_RMCHAR, &font).**

- parameter :

***osd\_fd*** : OSD device handler

***font*** : character index

**Examples:**

[cap\\_osd\\_string.c](#).

Definition at line [239](#) of file [vcap\\_osd.h](#).

```
#define FIOSDG_CHARMAP _IOR(VCAP_IOC_MAGIC, 10, fiosd_c
```

Use to get the bit map of characters

`ioctl(osd_fd, FIOSDG_CHARMAP, &charmap).`

- parameter :

`osd_fd` : OSD device handler

`charmap` : reference to `fiosd_charmap_t` structure

Definition at line [249](#) of file `vcap_osd.h`.

```
#define FIOSDG_HWINFO _IOR(VCAP_IOC_MAGIC, 11, fiosd_hw_
```

Use to get OSD hardware information

`ioctl(osd_fd, FIOSDG_HWINFO, &hwinfo).`

- parameter :

`osd_fd` : OSD device handler

`hwinfo` : reference to `fiosd_hw_info_t` structure

Definition at line [259](#) of file `vcap_osd.h`.

```
#define FIOSDS_BWINFOSET _IOR(VCAP_IOC_MAGIC, 12, int)
```

Use relative OSD window position and dimension setup control

`ioctl(osd_fd, FIOSDS_BWINFOSET, &enable).`

- parameter :

`osd_fd` : OSD device handler

`enable` : 0: disable, 1: enable

Definition at line [269](#) of file `vcap_osd.h`.

```
#define FIOSDS_FRAMEMODE _IOR(VCAP_IOC_MAGIC, 13, int)
```

Use to enable/disable OSD force frame type

**ioctl(osd\_fd, FIOSDS\_FRAMEMODE, &enable).**

- parameter :

**osd\_fd** : OSD device handler

**enable** : 0: disable, 1: enable

Definition at line [279](#) of file **vcap\_osd.h**.

```
#define FIOSDMASKS_ON _IOW(VCAP_IOC_MAGIC, 30, int)
```

Use to enable OSD mask window

**ioctl(osd\_fd, FIOSDMASKS\_ON, &windex).**

- parameter :

**osd\_fd** : OSD device handler

**windex** : specify to enable which OSD mask window, from 0~7

**Examples:**

[cap\\_osd\\_mask.c](#).

Definition at line [289](#) of file **vcap\_osd.h**.

```
#define FIOSDMASKS_OFF _IOW(VCAP_IOC_MAGIC, 31, int)
```

Use to disable OSD mask window

**ioctl(osd\_fd, FIOSDMASKS\_OFF, &windex).**

- parameter :

**osd\_fd** : OSD device handler

**windex** : specify to disable which OSD mask window, from 0~7

**Examples:**

[cap\\_osd\\_mask.c](#).

Definition at line [299](#) of file [vcap\\_osd.h](#).

```
#define FIOSDMASKS_TRANSPARENT _IOW(VCAP_IOC_MAGIC,
```

Use to setup OSD mask transparent level

`ioctl(osd_fd, FIOSDMASKS_TRANSPARENT, &mask_tran).`

- parameter :

*osd\_fd* : OSD device handler

*mask\_tran* : reference to fiosd\_transparent\_t structure

**Examples:**

[cap\\_osd\\_mask.c](#).

Definition at line [309](#) of file [vcap\\_osd.h](#).

```
#define FIOSDMASKS_WIN _IOW(VCAP_IOC_MAGIC, 33, fiosdma
```

Use to setup OSD mask window size and position

`ioctl(osd_fd, FIOSDMASKS_WIN, &win).`

- parameter :

*osd\_fd* : OSD device handler

*win* : reference to fiosmask\_win\_t structure

**Examples:**

[cap\\_osd\\_mask.c](#).

Definition at line [319](#) of file [vcap\\_osd.h](#).

```
#define FIOSDMASKS_BWINOFFSET _IOW(VCAP_IOC_MAGIC, 3
```

Use relative OSD mask window position and dimension setup control

**ioctl(osd\_fd, FIOSDMASKS\_BWINOFFSET, &enable).**

- parameter :  
**osd\_fd** : OSD device handler  
**enable** : 0: disable, 1: enable

Definition at line [329](#) of file [vcap\\_osd.h](#).

---

## Typedef Documentation

### **typedef struct fiosd\_win fiosd\_win\_t**

This structure defines for setting OSD window position and size.

### **typedef struct fiosd\_font\_info fiosd\_font\_info\_t**

This structure defines for setting OSD font feature.

If width and height smaller than fonts in OSD ram, the display font will be cut.

For example:

If the fonts size is 12x16 in sram, we set the **fiosd\_font\_info** as 6x8. The OSD will display the quarter font from top-left.

### **typedef struct fiosd\_palette fiosd\_palette\_t**

This structure defines for setting palette color.

### **typedef struct fiosd\_char fiosd\_char\_t**

This structure defines for add or replace a character in OSD ram.

The index 0 is special for FIOSDS\_RMCHAR.

### **typedef struct fiosd\_string fiosd\_string\_t**

This structure defines for setting display string.

The OSD will display string from fiosd\_string->start to fiosd\_string->start +

`strlen(fiosd_string->string)` in OSD display array. The string[128] is string index that depend on char bit map. While using multiple OSD windows at the same time, user has to manager string length in the common string buffer.

### **`typedef struct fiosd_transparent fiosd_transparent_t`**

This structure defines for setting transparent level.

### **`typedef struct fiosd_charmap fiosd_charmap_t`**

This structure defines for getting the bit map of characters.

### **`typedef struct fiosdmask_win fiosdmask_win_t`**

This structure defines for setting mask window postion and size.

### **`typedef struct fiosd_hw_info fiosd_hw_info_t`**

This structure defines for getting OSD hardware information.

[Main Page](#)

[Data Structures](#)

[Files](#)

[Examples](#)

# Examples

---

Here is a list of all examples:

- [`cap\_osd\_mask.c`](#)
  - [`cap\_osd\_string.c`](#)
- 

Generated on Thu Nov 25 2010 17:50:15 for GM8126/GM812X Capture IOCTL and API

Reference by [doxygen](#) 1.7.1

[Main Page](#)

[Data Structures](#)

[Files](#)

[Examples](#)

## cap\_osd\_mask.c

```
/**  
 * this sample code implement capture osd function  
, and record for 10 seconds.  
 * When you play this encode file, you can find os  
d string.  
 *  
 */  
  
#include <unistd.h>  
#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
#include <sys/types.h>  
#include <sys/stat.h>  
#include <fcntl.h>  
#include <sys/mman.h>  
#include <poll.h>  
#include <sys/ioctl.h>  
#include <sys/time.h>  
  
#include "dvr_common_api.h"  
#include "dvr_enc_api.h"  
#include "gmavi_api.h"  
#include "vcap_osd.h"  
  
int dvr_fd = 0;  
int enc_fd = 0;  
int video_fd[8] = { 0 };  
  
//test record  
unsigned char *bs_buf;  
HANDLE rec_file;
```

```
int avi_str_id;
int enc_buf_size;
struct pollfd rec_fds;
char file_name[128];

int bs_buf_snap_offset;
int encode_frame_count = 0;

dvr_enc_channel_param ch_param;
EncParam_Ext3 enc_param_ext = {0};
dvr_enc_control enc_ctrl;
FuncTag tag;

dvr_enc_channel_param user_rec_ch_setting =
{
{
    0,
    ENC_TYPE_FROM_CAPTURE,
    {352, 240},
    LVFRAME_EVEN_ODD,
    LVFRAME_FRAME_MODE,
    DMAORDER_PACKET,
    CAPSCALER_NOT_KEEP_RATIO,
    MCP_VIDEO_NTSC,
    CAPCOLOR_YUV422,
    { FALSE, FALSE, GM3DI_FIELD }
},
{
    DVR_ENC_EBST_ENABLE,
    0,
    ENC_TYPE_H264,
    FALSE,
    DVR_ENC_EBST_ENABLE,
    {352, 240},
    {ENC_INPUT_H2642D, 30, 1048576, 30, 25, 5
1, 1 , FALSE, {0, 0, 352, 240}},
    {SCALE_YUV422, SCALE_YUV422, SCALE_LINEAR,
```

```
    FALSE, FALSE, TRUE, 0 },
    {JCS_yuv420, 0, JENC_INPUT_MP42D, 70}

};

void do_record_start(int ch_num)
{
    memcpy(&ch_param, &user_rec_ch_setting, sizeof(ch_param));

    ch_param.main_bs.enc.ext_size = DVR_ENC_MAGIC_
ADD_VAL(sizeof(enc_param_ext));
    ch_param.main_bs.enc.pext_data = &enc_param_ex
t;

    enc_param_ext.feature_enable = 0;

    ioctl(enc_fd, DVR_ENC_SET_CHANNEL_PARAM, &ch_p
aram);

    ioctl(enc_fd, DVR_ENC_QUERY_OUTPUT_BUFFER_SIZE
, &enc_buf_size);

    ioctl(enc_fd, DVR_ENC_QUERY_OUTPUT_BUFFER_SNAP
_OFFSET, &bs_buf_snap_offset);

    bs_buf = (unsigned char*) mmap(NULL, enc_buf_s
ize, PROT_READ|PROT_WRITE,
                                MAP_SHAR
ED, enc_fd, 0);
//record start
    memset(&enc_ctrl, 0x0, sizeof(dvr_enc_control))
;
    enc_ctrl.command = ENC_START;
    enc_ctrl.stream = 0;
    ioctl(enc_fd, DVR_ENC_CONTROL, &enc_ctrl);
```

```
// set function tag parameter to dvr graph lev
el
    FN_RESET_TAG(&tag);
    FN_SET_REC_CH(&tag, ch_num);
    ioctl(dvr_fd, DVR_COMMON_APPLY, &tag);
}

void do_record_stop(int ch_num)
{
    //record stop
    memset(&enc_ctrl, 0x0, sizeof(dvr_enc_control));
    enc_ctrl.stream = 0;
    enc_ctrl.command = ENC_STOP;
    ioctl(enc_fd, DVR_ENC_CONTROL, &enc_ctrl);

    FN_RESET_TAG(&tag);
    FN_SET_REC_CH(&tag, ch_num);
    ioctl(dvr_fd, DVR_COMMON_APPLY, &tag);
    munmap((void*)bs_buf, enc_buf_size);
}

int osd_setting(void)
{
    fiosdmask_win_t win;
    fiosd_palette_t palette;
    fiosd_transparent_t tmp_transparent;
    int index = 0;
    int ret, i;
    int i_luminance, i_b_luminance;
    int i_cr, i_cb, i_b_cr, i_b_cb;
    char devicename[100];

    for (i = 0; i < 1; i++) {
        memset(devicename, 100, 0);
        sprintf(devicename, "/dev/fosd%d%d", i, 0)
```

```
;

    printf("open %s\n", devicename);
    video_fd[i] = open(devicename, O_RDWR);
    if (video_fd[i] < 0) {
        printf("fosd1 open fail!");
        exit(1);
    }

    printf("open device:%s successfully!\n", devicename);
    //foreground
    i_luminance = 120;
    i_cb = 128;
    i_cr = 128;
    //background
    i_b_luminance = 30;
    i_b_cb = 128;
    i_b_cr = 128;

    ret = ioctl(video_fd[i], FIOSDMASKS_OFF, &index);

    //transparent 0-0%, 1-50%, 2-75%, 3-100%
    tmp_transparent.windex = index;
    tmp_transparent.level = 0;
    ret = ioctl(video_fd[index], FIOSDMASKS_TRANSPARENT, &tmp_transparent);
    if (ret < 0) {
        printf("FIOSDMASKS_TRANSPARENT Fail!\n");
    }
    goto end;
}

win.windex = index;
win.x = 100;
win.y = 100;
win.width = 100;
```

```
        win.height = 100;
        win.color = 1;

        ret = ioctl(video_fd[index], FIOSDMASKS_WIN
, &win);
        if (ret < 0) {
            printf("FIOSDMASKS_WIN Fail!");
            goto end;
        }

        palette.index = 0;
        palette.y = i_luminance;
        palette.cb = i_cb;
        palette.cr = i_cr;
        ret = ioctl(video_fd[i], FIOSDS_PALTCOLOR,
&palette);
        if (ret < 0) {
            printf("FIOSDS_PALTCOLOR 0 Fail!");
            goto end;
        }
        palette.index = 1;
        palette.y = i_b_luminance;
        palette.cb = i_b_cb;
        palette.cr = i_b_cr;
        ret = ioctl(video_fd[i], FIOSDS_PALTCOLOR,
&palette);
        if (ret < 0) {
            printf("FIOSDS_PALTCOLOR 1 Fail!");
            goto end;
        }

        ret = ioctl(video_fd[i], FIOSDMASKS_ON, &i
ndex);
        if (ret < 0) {
            printf("FIOSDMASKS_ON Fail!");
            goto end;
        }
```

```
    }
end:
    return 0;
}

/***
 * @brief main function
 * @return 0 on success, !0 on error
 */
int main(int argc, char *argv[])
{
    int ret = 0, ch_num = 0;
    dvr_enc_queue_get    data;
    unsigned char *buf;
    int buf_size, i;
    struct timeval t1,t2;
    char tmp_str[128];

    dvr_fd = open("/dev/dvr_common", O_RDWR);      // 
open_dvr_common

    enc_fd = open("/dev/dvr_enc", O_RDWR);          // 
open_dvr_encode

    ret = osd_setting();
    if(ret < 0)
        goto end;

    do_record_start(ch_num);

    sprintf(file_name, "CH%d_video_%d", 0, 0);

    sprintf(tmp_str, "%s.h264", file_name);

    rec_file = fopen ( tmp_str , "wb+" );
gettimeofday(&t1, NULL);
```

```
while(1) {
    // prepare to select(or poll)
    rec_fds.fd = enc_fd;
    rec_fds.revents = 0;
    rec_fds.events = POLLIN_MAIN_BS;

    poll(&rec_fds, 1, 500);

    if (!(rec_fds.revents & POLLIN_MAIN_BS))
        continue;

    // get dataout buffer
    ret = ioctl(enc_fd, DVR_ENC_QUEUE_GET, &data);
    if(ret < 0)
        continue;

    encode_frame_count++;
    buf = bs_buf + data.bs.offset;
    buf_size = data.bs.length;

    fwrite (buf , 1 , buf_size , rec_file);
    fflush(rec_file);
    ioctl(enc_fd, DVR_ENC_QUEUE_PUT, &data);

    gettimeofday(&t2, NULL);

    if ((t2.tv_sec - t1.tv_sec) == 10) {
//< record for 10 seconds. then finish record.

        fclose(rec_file);
        break;
    }
}

do_record_stop(ch_num);
```

```
    printf("-----\n");
;
printf(" Record finish\n");
printf("-----\n");
;

ch_num = 0;
for (i = 0; i < 1; i++) {
    ret = ioctl(video_fd[i], FIOSDMASKS_OFF,
&ch_num);
    close(video_fd[i]);
}
end:
close(enc_fd);           //close_dvr_encode
close(dvr_fd);           //close_dvr_common

return 0;
}
```

[Main Page](#)

[Data Structures](#)

[Files](#)

[Examples](#)

## cap\_osd\_string.c

```
/**  
 * this sample code implement capture osd function  
, and record for 10 seconds.  
 * When you play this encode file, you can find os  
d string.  
 */  
  
#include <unistd.h>  
#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
#include <sys/types.h>  
#include <sys/stat.h>  
#include <fcntl.h>  
#include <sys/mman.h>  
#include <poll.h>  
#include <sys/ioctl.h>  
#include <sys/time.h>  
  
#include "dvr_common_api.h"  
#include "dvr_enc_api.h"  
#include "gmavi_api.h"  
#include "vcap_osd.h"  
  
int dvr_fd = 0;  
int enc_fd = 0;  
int video_fd[8] = { 0 };  
  
uint8_t space[36] = { 0x00, 0x00, 0x00, 0x00, 0x00  
, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00  
};
```

```
, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
};

//test record
unsigned char *bs_buf;
HANDLE rec_file;
int avi_str_id;
int enc_buf_size;
struct pollfd rec_fds;
char file_name[128];

int sub1_bs_buf_offset;
int sub2_bs_buf_offset;
int bs_buf_snap_offset;
int encode_frame_count = 0;

dvr_enc_channel_param ch_param;
EncParam_Ext3 enc_param_ext = {0};
dvr_enc_control enc_ctrl;
FuncTag tag;

dvr_enc_channel_param user_rec_ch_setting =
{
{
    0,
    ENC_TYPE_FROM_CAPTURE,
    {352, 240},
    LVFRAME_EVEN_ODD,
    LVFRAME_FRAME_MODE,
    DMAORDER_PACKET,
    CAPSCALER_NOT_KEEP_RATIO,
    MCP_VIDEO_NTSC,
    CAPCOLOR_YUV422,
    { FALSE, FALSE, GM3DI_FIELD }
},
{
```

```
    DVR_ENC_EBST_ENABLE,
    0,
    ENC_TYPE_H264,
    FALSE,
    DVR_ENC_EBST_ENABLE,
    {352, 240},
    {ENC_INPUT_H2642D, 30, 1048576, 30, 25, 5
1, 1 , FALSE, {0, 0, 352, 240}},
    {SCALE_YUV422, SCALE_YUV422, SCALE_LINEAR,
FALSE, FALSE, TRUE, 0 },
    {JCS_yuv420, 0, JENC_INPUT_MP42D, 70}

}

};

void do_record_start(int ch_num)
{
    memcpy(&ch_param, &user_rec_ch_setting, sizeof(ch_param));

    ch_param.main_bs.enc.ext_size = DVR_ENC_MAGIC_
ADD_VAL(sizeof(enc_param_ext));
    ch_param.main_bs.enc.pext_data = &enc_param_ex
t;

    enc_param_ext.feature_enable = 0;

    ioctl(enc_fd, DVR_ENC_SET_CHANNEL_PARAM, &ch_p
aram);

    ioctl(enc_fd, DVR_ENC_QUERY_OUTPUT_BUFFER_SIZE
, &enc_buf_size);

    ioctl(enc_fd, DVR_ENC_QUERY_OUTPUT_BUFFER_SNAP
_OFFSET, &bs_buf_snap_offset);

    bs_buf = (unsigned char*) mmap(NULL, enc_buf_s
```

```
ize, PROT_READ|PROT_WRITE,
MAP_SHARED);
ED, enc_fd, 0);
//record start
memset(&enc_ctrl, 0x0, sizeof(dvr_enc_control));
enc_ctrl.command = ENC_START;
enc_ctrl.stream = 0;
ioctl(enc_fd, DVR_ENC_CONTROL, &enc_ctrl);

// set function tag parameter to dvr graph level
FN_RESET_TAG(&tag);
FN_SET_REC_CH(&tag, ch_num);
ioctl(dvr_fd, DVR_COMMON_APPLY, &tag);
}

void do_record_stop(int ch_num)
{
//record stop
memset(&enc_ctrl, 0x0, sizeof(dvr_enc_control));
enc_ctrl.stream = 0;
enc_ctrl.command = ENC_STOP;
ioctl(enc_fd, DVR_ENC_CONTROL, &enc_ctrl);

FN_RESET_TAG(&tag);
FN_SET_REC_CH(&tag, ch_num);
ioctl(dvr_fd, DVR_COMMON_APPLY, &tag);
munmap((void*)bs_buf, enc_buf_size);
}

void initial_font(int channel)
{
fiosd_char_t ch;
//fiosd_charmap_t pr;
int ret;
```

```

char tmp;
//int i;

    tmp = 'A';           //take off 'A'
    ret = ioctl(video_fd[channel], FIOSDS_RMCHAR,
&tmp);

    ch.font = 'A';
    memcpy((void *)ch.fbitmap, (void *)space, sizeof(uint8_t) * 36);
    ioctl(video_fd[channel], FIOSDS_CHAR, &ch);
}

void set_osd_win_color(int osd_fd, int idx, int x,
int y, int w, int h, char *str, int start,
                uint8_t pal_fg, uint8_t pal
                _bg, uint8_t PY1, uint8_t PCb1, uint8_t PCr1, uint
                8_t PY2, uint8_t PCb2, uint8_t PCr2)
{
    fiosd_win_t win;
    fiosd_palette_t palette;
    fiosd_string_t string;
    fiosd_transparent_t transparent;
    int index;
    int ret;

    /* disable OSD windows */
    index = idx;
    ret = ioctl(osd_fd, FIOSDS_OFF, &index);
    if(ret < 0) {
        printf("FIOSDS_OFF Fail!");
        return;
    }

    /* set OSD window x,y postion and dimension */

    win.x      = x;

```

```
    win.y      = y;
    win.hdim   = w;
    win.vdim   = h;
    win.windex = idx;
    ret = ioctl(osd_fd, FIOSDS_WIN, &win);
    if(ret < 0) {
        printf("FIOSDS_WIN Fail!");
        return;
    }

    /* set OSD foreground color */
    palette.index = pal_fg;
    palette.y     = PY1;
    palette.cb    = PCb1;
    palette.cr    = PCr1;
    ret = ioctl(osd_fd, FIOSDS_PALTCOLOR, &palette
);
    if(ret < 0) {
        printf("FIOSDS_PALTCOLOR 0 Fail!");
        return;
    }

    /* set OSD backfrpund color */
    palette.index = pal_bg;
    palette.y     = PY2;
    palette.cb    = PCb2;
    palette.cr    = PCr2;
    ret = ioctl(osd_fd, FIOSDS_PALTCOLOR, &palette
);
    if(ret < 0) {
        printf("FIOSDS_PALTCOLOR 1 Fail!");
        return;
    }

    /* set OSD Transparent */
    transparent.level = FOSD_TRANSPARENT_100PERCE
NT;
```

```
    transparent.windex = index;
    ret = ioctl(osd_fd, FIOSDS_TRANSPARENT, &transparent);
    if(ret < 0) {
        printf("FIOSDMASKS_TRANSPARENT Fail!");
        return;
    }

/* set OSD window color and display string */

    string.windex    = index;
    string.start     = start;
    string.fg_color = pal_fg;
    string.bg_color = pal_bg;
    if(str)
        strcpy(&string.string[0], str);
    ret = ioctl(osd_fd, FIOSDS_STRING, &string);
    if(ret < 0) {
        printf("FIOSDS_STRING Fail!");
        return;
    }

/* enable OSD display */
    ret = ioctl(osd_fd, FIOSDS_ON, &index);
    if(ret < 0) {
        printf("FIOSDS_ON Fail!");
        return;
    }
}

int osd_setting(void)
{
    char str[256];
    int disp_idx, i;

    char devicename[100];
```

```
for (i = 0; i < 1; i++) {
    memset(devicename, 100, 0);
    sprintf(devicename, "/dev/fosd%d%d", i, 0)
};

printf("open %s\n", devicename);
video_fd[i] = open(devicename, O_RDWR);
if (video_fd[i] < 0) {
    printf("fosd1 open fail!");
    exit(1);
}
initial_font(i);
printf("open device:%s successfully!\n", devicename);

strcpy(str, "WIN 0");
disp_idx = 32;
str[32] = 0;
set_osd_win_color(video_fd[i], 0, 10, 50,
5, 1, str, disp_idx, 0, 1,
200, 200, 200, 60, 60, 60);
0);

strcpy(str, "WIN 1");
disp_idx = 96;
str[32] = 0;
set_osd_win_color(video_fd[i], 1, 10, 150,
5, 1, str, disp_idx, 1, 0,
200, 200, 200, 60, 60, 60);
0);

strcpy(str, "WIN 2");
disp_idx = 0;
str[32] = 0;
set_osd_win_color(video_fd[i], 2, 100, 50,
5, 1, str, disp_idx, 0, 1,
200, 200, 200, 60, 60, 60);
0);
```

```

        strcpy(str, "WIN 3");
        disp_idx = 64;
        str[32] = 0;
        set_osd_win_color(video_fd[i], 3, 100, 150
, 5, 1, str, disp_idx, 1, 0,
                           200, 200, 200, 60, 60, 6
0);
    }
    return 0;
}

/**
 * @brief main function
 * @return 0 on success, !0 on error
 */
int main(int argc, char *argv[])
{
    int ret = 0, ch_num = 0;
    dvr_enc_queue_get    data;
    unsigned char *buf;
    int buf_size, i;
    struct timeval t1,t2;
    char tmp_str[128];

    dvr_fd = open("/dev/dvr_common", O_RDWR);      // 
open_dvr_common

    enc_fd = open("/dev/dvr_enc", O_RDWR);         // 
open_dvr_encode

    ret = osd_setting();
    if(ret < 0)
        goto end;

    do_record_start(ch_num);
}

```

```
 sprintf(file_name, "CH%d_video_%d", 0, 0);

 sprintf(tmp_str, "%s.h264", file_name);

 rec_file = fopen ( tmp_str , "wb+" );
 gettimeofday(&t1, NULL);

 while(1) {
    // prepare to select(or poll)
    rec_fds.fd = enc_fd;
    rec_fds.revents = 0;
    rec_fds.events = POLLIN_MAIN_BS;

    poll(&rec_fds, 1, 500);

    if (!(rec_fds.revents & POLLIN_MAIN_BS))
        continue;

    // get dataout buffer
    ret = ioctl(enc_fd, DVR_ENC_QUEUE_GET, &data);
    if(ret < 0)
        continue;

    encode_frame_count++;
    buf = bs_buf + data.bs.offset;
    buf_size = data.bs.length;

    fwrite (buf , 1 , buf_size , rec_file);
    fflush(rec_file);
    ioctl(enc_fd, DVR_ENC_QUEUE_PUT, &data);

    gettimeofday(&t2, NULL);

    if ((t2.tv_sec - t1.tv_sec) == 10) {
//< record for 10 seconds. then finish record.
```

```
        fclose(rec_file);
        break;
    }
}

do_record_stop(ch_num);

printf("-----\n");
;
printf(" Record finish\n");
printf("-----\n");
;

ch_num = 0;
for (i = 0; i < 1; i++) {
    ret = ioctl(video_fd[i], FIOSDS_OFF, &ch_
num);
    close(video_fd[i]);
}
end:
close(enc_fd);           //close_dvr_encode
close(dvr_fd);           //close_dvr_common

return 0;
}
```

Main Page	Data Structures	Files	Examples
File List	Globals		
All	Typedefs	Defines	
f h m o v w x y			

Here is a list of all functions, variables, defines, enums, and typedefs with links to the files they belong to:

### - f -

- fiosd\_char\_t : [vcap\\_osd.h](#)
- fiosd\_charmap\_t : [vcap\\_osd.h](#)
- fiosd\_font\_info\_t : [vcap\\_osd.h](#)
- fiosd\_hw\_info\_t : [vcap\\_osd.h](#)
- fiosd\_palette\_t : [vcap\\_osd.h](#)
- fiosd\_string\_t : [vcap\\_osd.h](#)
- fiosd\_transparent\_t : [vcap\\_osd.h](#)
- fiosd\_win\_t : [vcap\\_osd.h](#)
- FIOSDG\_CHARMAP : [vcap\\_osd.h](#)
- FIOSDG\_HWINFO : [vcap\\_osd.h](#)
- fiosdmask\_win\_t : [vcap\\_osd.h](#)
- FIOSDMASKS\_BWINFOFFSET : [vcap\\_osd.h](#)
- FIOSDMASKS\_OFF : [vcap\\_osd.h](#)
- FIOSDMASKS\_ON : [vcap\\_osd.h](#)
- FIOSDMASKS\_TRANSPARENT : [vcap\\_osd.h](#)
- FIOSDMASKS\_WIN : [vcap\\_osd.h](#)
- FIOSDS\_BWINFOFFSET : [vcap\\_osd.h](#)
- FIOSDS\_CHAR : [vcap\\_osd.h](#)
- FIOSDS\_FONTSETTING : [vcap\\_osd.h](#)
- FIOSDS\_FRAMEMODE : [vcap\\_osd.h](#)
- FIOSDS\_OFF : [vcap\\_osd.h](#)
- FIOSDS\_ON : [vcap\\_osd.h](#)
- FIOSDS\_PALTCOLOR : [vcap\\_osd.h](#)
- FIOSDS\_RMCHAR : [vcap\\_osd.h](#)
- FIOSDS\_STRING : [vcap\\_osd.h](#)
- FIOSDS\_TRANSPARENT : [vcap\\_osd.h](#)
- FIOSDS\_WIN : [vcap\\_osd.h](#)

- FOSD\_TRANSPARENT\_0PERCENT : [vcap\\_osd.h](#)
- FOSD\_TRANSPARENT\_100PERCENT : [vcap\\_osd.h](#)
- FOSD\_TRANSPARENT\_50PERCENT : [vcap\\_osd.h](#)
- FOSD\_TRANSPARENT\_75PERCENT : [vcap\\_osd.h](#)

- **h** -

- H2OFFSET : [vcap\\_osd.h](#)

- **m** -

- MAX\_TARGET : [vcap\\_osd.h](#)

- **o** -

- OFFSET2H : [vcap\\_osd.h](#)
- OFFSET2W : [vcap\\_osd.h](#)
- OFFSET2X : [vcap\\_osd.h](#)
- OFFSET2Y : [vcap\\_osd.h](#)

- **v** -

- VCAP\_IOC\_MAGIC : [vcap\\_osd.h](#)

- **w** -

- W2OFFSET : [vcap\\_osd.h](#)

- **x** -

- X2OFFSET : [vcap\\_osd.h](#)

- **y** -

- Y2OFFSET : [vcap\\_osd.h](#)

[Main Page](#)[Data Structures](#)[Files](#)[Examples](#)[File List](#)[Globals](#)

## source/vcap\_osd.h

Go to the documentation of this file.

```
00001 /** @file vcap_osd.h
00002 *
00003 * @brief This file provides the capture OSD and Mask related function calls or IOCTLs
00004 *
00005 * @author Francis Huang
00006 * @version 0.1.0
00007 * @date 2010/09/28
00008 *
00009 */
00010
00011 /** @mainpage GM8126/GM812X Capture IOCTL and API Reference Documentation
00012 *
00013 * @n vcap_osd.h describe the capture OSD and Mask related function calls or IOCTLs
00014 */
00015
00016 /**
00017 * @example cap_osd_string.c
00018 * @example cap_osd_mask.c
00019 */
00020
00021 #ifndef _VCAP_OSD_H_
00022 #define _VCAP_OSD_H_
00023 #include <linux/ioctl.h>
00024 #include <linux/videodev.h>
00025
00026 #ifndef __KERNEL__
00027 #include <stdint.h>
00028 #endif
00029
```

```
00030 #define MAX_TARGET 10000
00031
00032 #define X20OFFSET(base_w, x) (MAX_TARGET
* x / base_w)
00033 #define Y20OFFSET(base_h, y) (MAX_TARGET
* y / base_h)
00034 #define W20OFFSET(base_w, w) (MAX_TARGET
* w / base_w)
00035 #define H20OFFSET(base_h, h) (MAX_TARGET
* h / base_h)
00036
00037 #define OFFSET2X(base_w, off_x) (off_x * base_w / MAX_TARGET)
00038 #define OFFSET2Y(base_h, off_y) (off_y * base_h / MAX_TARGET)
00039 #define OFFSET2W(base_w, off_w) (off_w * base_w / MAX_TARGET)
00040 #define OFFSET2H(base_h, off_h) (off_h * base_h / MAX_TARGET)
00041
00042 /**
00043 * @brief This structure defines for setting
00044 * OSD window position and size.
00045 */
00046 typedef struct fiosd_win {
00047     uint32_t x:16;           ///< left position of OSD window
00048     uint32_t y:16;           ///< top position of OSD window
00049     uint32_t hdim:16;        ///< the dimension width of OSD window
00050     uint32_t vdim:16;        ///< the dimension height of OSD window
00051     uint8_t windex;          ///< the OSD window index
00052 }
```

```
00053 /**
00054  * @brief This structure defines for setting
00055  * @n      OSD font feature.
00056  * @n      If width and height smaller than f
00057  * @n      onts in OSD ram, the display font will be cut.
00058  * @n      For example:
00059  * @n      If the fonts size is 12x16 in sram
00060  * @n      , we set the fiosd_font_info as 6x8.
00061  * @n      The OSD will display the quarter f
00062  * @n      ont from top-left.
00063  */
00064 typedef struct fiosd_font_info {
00065     uint32_t width:8;           ///< the font wi
00066     dth that will be display
00067     uint32_t height:8;         ///< the font he
00068     ight that will be display
00069     uint32_t row_space:8;      ///< the font ro
00070     w space
00071     uint32_t col_space:8;      ///< the font co
00072     lumn space
00073     uint8_t windex;          ///< the OSD win
00074     dow index
00075 } fiosd_font_info_t;
00076
00077 /**
00078  * @brief This structure defines for setting
00079  * @n      palette color.
00080  */
00081 typedef struct fiosd_palette {
00082     uint8_t index;            ///< the palette
00083     color index. The value is from 0 to 6
00084     uint8_t y;                ///< the color o
00085     f Y
00086     uint8_t cb;               ///< the color o
00087     f Cb
00088     uint8_t cr;               ///< the color o
```

```
f Cr
00077 } fiosd_palette_t;
00078
00079 /**
00080 * @brief This structure defines for add or
replace a character in OSD ram.
00081 * @n
00082 * @n      The index 0 is special for FIOSDS_
RMCHAR.
00083 */
00084 typedef struct fiosd_char {
00085     char font;           // < the font index that you want to setting or replacing
00086     uint8_t fbitmap[36]; // < the font bitmap
00087 } fiosd_char_t;
00088
00089 /**
00090 * @brief This structure defines for setting
display string.
00091 * @n
00092 * @n      The OSD will display string from f
iosd_string->start to fiosd_string->start +
00093 * @n      strlen(fiosd_string->string) in OS
D display array. The string[128] is string
00094 * @n      index that depend on char bit map.
While using multiple OSD windows at the same
00095 * @n      time, user has to manager string length in the common string buffer.
00096 */
00097 typedef struct fiosd_string {
00098     uint8_t windex;        // < the OSD window index
00099     uint8_t start;         // < the string start index in OSD display array. 0-255
00100    uint8_t fg_color;      // < the forward color index that assign by FIOSDS_PALTCOLOR
```

```
00101     uint8_t bg_color;           //;< the backward  
d color index that assign by FIOSDS_PALTCOLOR  
00102     char string[128];          //;< the string  
that will be displayed  
00103 } fiosd_string_t;  
00104  
00105 /**  
00106 * @brief This structure defines for setting  
transparent level.  
00107 */  
00108 typedef struct fiosd_transparent {  
00109     uint8_t windex;             //;< the OSD window index  
00110     uint8_t level;            //;< the transparent level 0, 50, 75, 100 percent  
  
00111 #define FOSD_TRANSPARENT_0PERCENT    0  
00112 #define FOSD_TRANSPARENT_50PERCENT   1  
00113 #define FOSD_TRANSPARENT_75PERCENT   2  
00114 #define FOSD_TRANSPARENT_100PERCENT  3  
00115 } fiosd_transparent_t;  
00116  
00117 /**  
00118 * @brief This structure defines for getting  
the bit map of characters.  
00119 */  
00120 typedef struct fiosd_charmap {  
00121     uint32_t map[8];           //;< the bitmap  
of fonts that store in OSD ram  
00122 } fiosd_charmap_t;  
00123  
00124 /**  
00125 * @brief This structure defines for setting  
mask window position and size  
00126 */  
00127 typedef struct fiosdmask_win {  
00128     uint32_t x:16;            //;< left position
```

```
on of mask window
00129     uint32_t y:16;           ///< top position of mask window
00130     uint32_t width:16;        ///< the dimension width of mask window
00131     uint32_t height:16;       ///< the dimension height of mask window
00132     uint8_t windex;          ///< the mask window index
00133     uint8_t color;          ///< the mask color index that assign by FIOSDS_PALTCOLOR
00134 } fiosdmask_win_t;
00135
00136 /**
00137 * @brief This structure defines for getting OSD hardware information.
00138 */
00139 typedef struct fiosd_hw_info {
00140     int MaxFontNum;           ///< max OSD font number
00141     int MaxDispNum;          ///< max OSD display number
00142 } fiosd_hw_info_t;
00143
00144 /* Use 'f' as magic number */
00145 #define VCAP_IOC_MAGIC 'f'
00146
00147 /*
00148 * IOCTL
00149 */
00150
00151 /**
00152 * @brief Use to enable OSD window
00153 * @n
00154 * @n @b ioctl(osd_fd, FIOSDS_ON, &windex)
00155 * @arg parameter :
```

```
00156 * @n @b @e osd_fd : OSD device handler
00157 * @n @b @e windex : specify to enable which
    OSD window, from 0~3
00158 */
00159 #define FIOSDS_ON           _IOW(VCAP_IO
C_MAGIC, 1, int)
00160
00161 /**
00162 * @brief Use to disable OSD window
00163 * @n
00164 * @n @b ioctl(osd_fd, FIOSDS_OFF, &windex)
00165 * @arg parameter :
00166 * @n @b @e osd_fd : OSD device handler
00167 * @n @b @e windex : specify to disable which
    OSD window, from 0~3
00168 */
00169 #define FIOSDS_OFF          _IOW(VCAP_IO
C_MAGIC, 2, int)
00170
00171 /**
00172 * @brief Use to setup OSD window size and
    position
00173 * @n
00174 * @n @b ioctl(osd_fd, FIOSDS_WIN, &win)
00175 * @arg parameter :
00176 * @n @b @e osd_fd : OSD device handler
00177 * @n @b @e win : reference to fiosd_win_t s
    tructure
00178 */
00179 #define FIOSDS_WIN          _IOW(VCAP_IO
C_MAGIC, 3, fiosd_win_t)
00180
00181 /**
00182 * @brief Use to setup OSD font feature
00183 * @n
00184 * @n @b ioctl(osd_fd, FIOSDS_FONTSETTING
```

```
, &font_info)
00185 * @arg      parameter :
00186 * @n @b @e osd_fd : OSD device handler
00187 * @n @b @e font_info : reference to fiosd_f
ont_info_t structure
00188 */
00189 #define FIOSDS_FONTSETTING      _IOW(VCAP_IO
C_MAGIC, 4, fiosd_font_info_t)
00190
00191 /**
00192 * @brief    Use to setup OSD font transparent level
00193 * @n
00194 * @n @b     ioctl(osd_fd, FIOSDS_TRANSPARENT
, &font_tran)
00195 * @arg      parameter :
00196 * @n @b @e osd_fd : OSD device handler
00197 * @n @b @e font_tran : reference to fiosd_t
ransparent_t structure
00198 */
00199 #define FIOSDS_TRANSPARENT      _IOW(VCAP_IO
C_MAGIC, 5, fiosd_transparent_t)
00200
00201 /**
00202 * @brief    Use to add or replace a character in OSD ram
00203 * @n
00204 * @n @b     ioctl(osd_fd, FIOSDS_CHAR, &font
_char)
00205 * @arg      parameter :
00206 * @n @b @e osd_fd : OSD device handler
00207 * @n @b @e font_char : reference to fiosd_c
har_t structure
00208 */
00209 #define FIOSDS_CHAR            _IOW(VCAP_IO
C_MAGIC, 6, fiosd_char_t)
00210
```

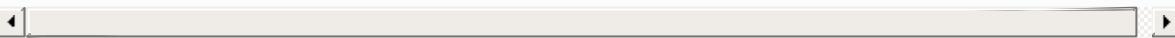
```
00211 /**
00212  * @brief    Use to setup OSD display string
00213  * @n
00214  * @n @b      ioctl(osd_fd, FIOSDS_STRING, &fo
00215  * @arg      parameter :
00216  * @n @b @e osd_fd : OSD device handler
00217  * @n @b @e font_string : reference to fiosd
00218 _string_t structure
00219 */
00220 #define FIOSDS_STRING           _IOW(VCAP_IO
C_MAGIC, 7, fiosd_string_t)
00221 /**
00222  * @brief    Use to setup OSD palette color
00223  * @n
00224  * @n @b      ioctl(osd_fd, FIOSDS_PALTCOLOR,
00225  * @arg      parameter :
00226  * @n @b @e osd_fd : OSD device handler
00227  * @n @b @e palette : reference to fiosd_pa
00228 _lette_t structure
00229 */
00230 #define FIOSDS_PALTCOLOR        _IOW(VCAP_IO
C_MAGIC, 8, fiosd_palette_t)
00231 /**
00232  * @brief    Use to remove a character from O
00233 SD ram
00234  * @n
00235  * @n @b      ioctl(osd_fd, FIOSDS_RMCHAR, &fo
00236 nt)
00237  * @arg      parameter :
00238  * @n @b @e osd_fd : OSD device handler
00239  * @n @b @e font : character index
00240 */
00241 #define FIOSDS_RMCHAR          _IOW(VCAP_IO
```

```
C_MAGIC, 9, char)
00240
00241 /**
00242 * @brief Use to get the bit map of characters
00243 * @n
00244 * @n @b ioctl(osd_fd, FIOSDG_CHARMAP, &charmap)
00245 * @arg parameter :
00246 * @n @b @e osd_fd : OSD device handler
00247 * @n @b @e charmap : reference to fiosd_charmap_t structure
00248 */
00249 #define FIOSDG_CHARMAP _IOR(VCAP_IO
C_MAGIC, 10, fiosd_charmap_t)
00250
00251 /**
00252 * @brief Use to get OSD hardware information
00253 * @n
00254 * @n @b ioctl(osd_fd, FIOSDG_HWINFO, &hwinfo)
00255 * @arg parameter :
00256 * @n @b @e osd_fd : OSD device handler
00257 * @n @b @e hwinfo : reference to fiosd_hw_info_t structure
00258 */
00259 #define FIOSDG_HWINFO _IOR(VCAP_IO
C_MAGIC, 11, fiosd_hw_info_t)
00260
00261 /**
00262 * @brief Use relative OSD window position and dimension setup control
00263 * @n
00264 * @n @b ioctl(osd_fd, FIOSDS_BWINOFFSET,
&enable)
00265 * @arg parameter :
```

```
00266 * @n @b @e osd_fd : OSD device handler
00267 * @n @b @e enable : 0: disable, 1: enable
00268 */
00269 #define FIOSDS_BWINOFFSET           _IOR(VCAP_IO
C_MAGIC, 12, int)
00270
00271 /**
00272 * @brief    Use to enable/disable OSD force
frame type
00273 * @n
00274 * @n @b      ioctl(osd_fd, FIOSDS_FRAMEMODE,
&enable)
00275 * @arg      parameter :
00276 * @n @b @e osd_fd : OSD device handler
00277 * @n @b @e enable : 0: disable, 1: enable
00278 */
00279 #define FIOSDS_FRAMEMODE           _IOR(VCAP_IO
C_MAGIC, 13, int)
00280
00281 /**
00282 * @brief    Use to enable OSD mask window
00283 * @n
00284 * @n @b      ioctl(osd_fd, FIOSDMASKS_ON, &wi
ndex)
00285 * @arg      parameter :
00286 * @n @b @e osd_fd : OSD device handler
00287 * @n @b @e windex : specify to enable which
OSD mask window, from 0~7
00288 */
00289 #define FIOSDMASKS_ON            _IOW(VCAP_IO
C_MAGIC, 30, int)
00290
00291 /**
00292 * @brief    Use to disable OSD mask window
00293 * @n
00294 * @n @b      ioctl(osd_fd, FIOSDMASKS_OFF, &w
index)
```

```
00295 * @arg      parameter :
00296 * @n @b @e osd_fd : OSD device handler
00297 * @n @b @e windex : specify to disable which OSD mask window, from 0~7
00298 */
00299 #define FIOSDMASKS_OFF           _IOW(VCAP_IO
C_MAGIC, 31, int)
00300
00301 /**
00302 * @brief    Use to setup OSD mask transparent level
00303 * @n
00304 * @n @b     ioctl(osd_fd, FIOSDMASKS_TRANSPARENT,
&mask_tran)
00305 * @arg      parameter :
00306 * @n @b @e osd_fd : OSD device handler
00307 * @n @b @e mask_tran : reference to fiosd_t
transparent_t structure
00308 */
00309 #define FIOSDMASKS_TRANSPARENT _IOW(VCAP_IO
C_MAGIC, 32, fiosd_transparent_t)
00310
00311 /**
00312 * @brief    Use to setup OSD mask window size and position
00313 * @n
00314 * @n @b     ioctl(osd_fd, FIOSDMASKS_WIN, &win)
00315 * @arg      parameter :
00316 * @n @b @e osd_fd : OSD device handler
00317 * @n @b @e win : reference to fiosdmask_win_t
structure
00318 */
00319 #define FIOSDMASKS_WIN           _IOW(VCAP_IO
C_MAGIC, 33, fiosdmask_win_t)
00320
00321 /**
```

```
00322 * @brief Use relative OSD mask window position and dimension setup control
00323 * @n
00324 * @n @b ioctl(osd_fd, FIOSDMASKS_BWINOFF
SET, &enable)
00325 * @arg parameter :
00326 * @n @b @e osd_fd : OSD device handler
00327 * @n @b @e enable : 0: disable, 1: enable
00328 */
00329 #define FIOSDMASKS_BWINOFFSET _IOW(VCAP_IO
C_MAGIC, 34, int)
00330 #endif /* _VCAP OSD_H_ */
```



Generated on Thu Nov 25 2010 17:50:15 for GM8126/GM812X Capture IOCTL and API

Reference by [doxygen](#) 1.7.1

Main Page	Data Structures	Files	Examples									
Data Structures	Data Fields											
All	Variables											
b	c	f	h	i	l	m	r	s	v	w	x	y

- b -

- bg\_color : **fiosd\_string**

- c -

- cb : **fiosd\_palette**
- col\_space : **fiosd\_font\_info**
- color : **fiosdmask\_win**
- cr : **fiosd\_palette**

- f -

- fbitmap : **fiosd\_char**
- fg\_color : **fiosd\_string**
- font : **fiosd\_char**

- h -

- hdim : **fiosd\_win**
- height : **fiosdmask\_win** , **fiosd\_font\_info**

- i -

- index : **fiosd\_palette**

- l -

- level : **fiosd\_transparent**

- m -

- map : **fiosd\_charmap**
- MaxDispNum : **fiosd\_hw\_info**
- MaxFontNum : **fiosd\_hw\_info**

- **r** -

- row\_space : **fiosd\_font\_info**

- **s** -

- start : **fiosd\_string**
- string : **fiosd\_string**

- **v** -

- vdim : **fiosd\_win**

- **w** -

- width : **fiosd\_font\_info** , **fiosdmask\_win**
- windex : **fiosd\_transparent** , **fiosd\_string** , **fiosd\_font\_info** ,  
**fiosd\_win** , **fiosdmask\_win**

- **x** -

- x : **fiosd\_win** , **fiosdmask\_win**

- **y** -

- y : **fiosd\_win** , **fiosdmask\_win** , **fiosd\_palette**

- fiosd\_char\_t : [vcap\\_osd.h](#)
- fiosd\_charmap\_t : [vcap\\_osd.h](#)
- fiosd\_font\_info\_t : [vcap\\_osd.h](#)
- fiosd\_hw\_info\_t : [vcap\\_osd.h](#)
- fiosd\_palette\_t : [vcap\\_osd.h](#)
- fiosd\_string\_t : [vcap\\_osd.h](#)
- fiosd\_transparent\_t : [vcap\\_osd.h](#)
- fiosd\_win\_t : [vcap\\_osd.h](#)
- fiosdmask\_win\_t : [vcap\\_osd.h](#)

Main Page	Data Structures	Files	Examples
File List	Globals		
All	Typedefs	Defines	
f h m o v w x y			

## - f -

- FIOSDG\_CHARMAP : [vcap\\_osd.h](#)
- FIOSDG\_HWINFO : [vcap\\_osd.h](#)
- FIOSDMASKS\_BWINOFFSET : [vcap\\_osd.h](#)
- FIOSDMASKS\_OFF : [vcap\\_osd.h](#)
- FIOSDMASKS\_ON : [vcap\\_osd.h](#)
- FIOSDMASKS\_TRANSPARENT : [vcap\\_osd.h](#)
- FIOSDMASKS\_WIN : [vcap\\_osd.h](#)
- FIOSDS\_BWINOFFSET : [vcap\\_osd.h](#)
- FIOSDS\_CHAR : [vcap\\_osd.h](#)
- FIOSDS\_FONTSETTING : [vcap\\_osd.h](#)
- FIOSDS\_FRAMEMODE : [vcap\\_osd.h](#)
- FIOSDS\_OFF : [vcap\\_osd.h](#)
- FIOSDS\_ON : [vcap\\_osd.h](#)
- FIOSDS\_PALTCOLOR : [vcap\\_osd.h](#)
- FIOSDS\_RMCHAR : [vcap\\_osd.h](#)
- FIOSDS\_STRING : [vcap\\_osd.h](#)
- FIOSDS\_TRANSPARENT : [vcap\\_osd.h](#)
- FIOSDS\_WIN : [vcap\\_osd.h](#)
- FOSD\_TRANSPARENT\_0PERCENT : [vcap\\_osd.h](#)
- FOSD\_TRANSPARENT\_100PERCENT : [vcap\\_osd.h](#)
- FOSD\_TRANSPARENT\_50PERCENT : [vcap\\_osd.h](#)
- FOSD\_TRANSPARENT\_75PERCENT : [vcap\\_osd.h](#)

## - h -

- H2OFFSET : [vcap\\_osd.h](#)

## - m -

- MAX\_TARGET : [vcap\\_osd.h](#)

- o -

- OFFSET2H : [vcap\\_osd.h](#)
- OFFSET2W : [vcap\\_osd.h](#)
- OFFSET2X : [vcap\\_osd.h](#)
- OFFSET2Y : [vcap\\_osd.h](#)

- v -

- VCAP\_IOC\_MAGIC : [vcap\\_osd.h](#)

- w -

- W2OFFSET : [vcap\\_osd.h](#)

- x -

- X2OFFSET : [vcap\\_osd.h](#)

- y -

- Y2OFFSET : [vcap\\_osd.h](#)