SENSORAY CO., INC.

516/616 MPEG Frame Grabbers

Programmer's Manual

May 21, 2004

7313 SW Tech Center Dr.
Tigard, OR 97223
Phone 503.684.8073 • Fax 503.684.8164
sales@sensoray.com
www.sensoray.com

1. Introduction *4*
2. Software Components *5*
2.1. Installation Procedure **7**
3. Building an Application with SX16 Library *8*
4. Functions Reference *8*
4.1. Windows Specific Functions **8**
X16_DLLOpen 8
X16_DLLClose 9
4.2. Library Functions **10**
4.2.1. System Functions 10
X16_open 10
X16_register 10
X16_unregister 11
X16_ getboardinfo 11
X16_close 11
4.2.2. Video Functions 12
X16_set_input 12
X16_set_input_type 12
X16_set_input_system 12
X16_get_input_status 13
X16_set_output_system 13
X16_set_bitrate 14
X16_set_picsize 14
X16_set_n 15
X16_set_m 15
X16_set_mux 15
X16_set_vbr 16
X16_read 16
X16_write 17
X16_stop 17
X16_pause 18
X16_resume 18
X16_set_brightness 18
X16_set_contrast 19
X16_set_saturation 19
X16_set_colorbars 19
4.2.3. Audio Functions 21
X16_set_volume 21
X16_set_mute 21
X16_set_audio_pars 21
4.2.4. Capture Functions 24
X16_prepare_capture 24
X16_get_capture_status 24

X16_capture_start 24
X16_close_capture 25
4.2.5. Overlay Functions 26
X16_overlay_get 26
X16_overlay_setmode 27
X16_overlay_setpos 27
X16_overlay_setcolors 27
X16_overlay_write 28
4.2.6. Digital I/O Functions 29
X16_read_gpio 29
X16_write_gpio 29

1. Introduction

The Sensoray Models 516 and 616 is an MPEG video encoder decoder board. Some of the features include:

General

- Real time MPEG-2 and MPEG-1 video encoder and decoder
- Support for variable bit rate and constant bit rate
- IPB pictures to 15Mbps for constant bit rate and 10Mbps for variable bit rate
- Supports multiple resolutions (704x480, 640x480, 352x240, etc.)
- Support for NTSC, PAL
- During encoding and standby, video input is fed to output for easy adjustment
- Onboard audio CODEC

Video encoder

- Generates 13818 (MPEG-2) and 11172 (MPEG-1) compliant elementary streams (ES)
- Operates up to 30 frames per second
- Selectable bit rate

Video decoder

- Decodes both MPEG-1 and MPEG-2 streams
- Horizontal and vertical scaling

Video capture (616 board only)

- Supports small (up to 1/2x1/2 of original size) uncompressed image capture.

Video overlay (616 board only)

- Supports 256x32 user defined monochrome picture above recorded video.

2. Software Components

The software contains the following files:

Windows 98, ME, 2000, XP

Filename	Function
sm2288.h	Header: hardware-depended constants
sm2288f.h	Header: function prototypes
sm2288.dll	Library functions (DLL)
sxdrv98.sys	The driver
sx16.inf	Windows INF to register the driver
x16demo.c	Demo application
x16app.c	Some common components to connect to the DLL
x16demo.h	Header file for the demo
x16demo.rc	Resource file for the demo
*.ico, *.bmp	Pictures for the demo

Table 1. Software Components (Windows WDM)

Windows NT 4

Filename	Function
sm2288.h	Header: hardware-depended constants
sm2288f.h	Header: function prototypes
sm2288.dll	Library functions (DLL)
sx16.ini	List of acceptable base addresses (for 516 board only)
sxdrvnt.sys	The driver
setupnt.bat	Setup batch file
regini.exe	Driver registration program
sxdriver.ini	INI file to register the driver
x16demo.c	Demo application
x16app.c	Some common components to connect to the DLL
x16demo.h	Header file for the demo
x16demo.rc	Resource file for the demo
*.ico, *.bmp	Pictures for the demo

Table 2. Software Components (Windows NT)

Windows CE

Filename	Function
sm2288.h	Header: hardware-depended constants
sm2288f.h	Header: function prototypes
x16.dll	The driver
x16demo.c	Demo application
x16app.c	Some common components to connect to the driver
x16demo.h	Header file for the demo
x16demo.rc	Resource file for the demo
*.ico, *.bmp	Pictures for the demo

Table 3. Software Components (Windows CE)

Linux

Filename	Function
sm2288.h	Header: hardware-depended constants
sm2288f.h	Header: function prototypes
sx16lib.o	Library functions
x16dat1.hex	Hardware data files
x16dat2.hex	
x16dat3.hex	
x16dat4.hex	
x16dat5.hex	
x16dat6.hex	
sx16.ini	List of acceptable base addresses (for 516 board only)
sxdrvu.o	The driver
mknode	Script to register the driver
x16demo.c	Demo application
makefile	The rule to build the demo

Table 4. Software Components (Linux)

QNX

Filename	Function
sm2288.h	Header: hardware-depended constants
sm2288f.h	Header: function prototypes
sm2288.o	Library functions
x16dat1.hex	Hardware data files
x16dat2.hex	
x16dat3.hex	
x16dat4.hex	
x16dat5.hex	
x16dat6.hex	
sx16.ini	List of acceptable base addresses (for 516 board only)
sxdrvq.o	The driver
x16demo.c	Demo application
makefile	The rule to build the demo

Table 5. Software Components (QNX)

2.1. Installation Procedure

Windows 98, ME, 2000, XP

Register the driver with the "Device Manager" utility (Add New Hardware).

To register a 516 board you must select the IO base address in this dialog:

- Select "No, I want to select the hardware from a list".
- Select the type of hardware: "Other devices"
- Click "Have Disk"
- Browse the location of sx16.inf file
- Select "Sensoray Model 516 Frame Grabber"
- You will see the list of resources (Input/Output Range), acceptable for the device. Select one and set the same base address on the board (see 516 board manual).

Windows NT 4

Run *setupnt.bat*. Restart Windows.

Windows CE

Copy the driver file to your Windows System32\Drivers directory.

Linux

Run *mknode* to register the driver. Type: *insmod sxdrvu.o* to run the driver.

QNX

No special actions required.

3. Building an Application with SX16 Library

Windows, Windows CE

Build your application together with x16app.c, including sm2288.h and sm2288f.h.

Linux

Compile your application including sm2288.h and sm2288f.h.
Link your object file together with sx16lib.o

QNX

Compile your application including sm2288.h and sm2288f.h.
Link your object file together with sm2288.o and sxdrvq.o.

Please refer to the sample source code for an example of building an application with SX16 Library.

4. Functions Reference

The SX16 Library is designed to provide the application developer with full control over the frame grabber. All special data types used by the Library are defined in sm2288.h. The sample application illustrates the use of most of the functions and allows building a custom application within minutes.

4.1. Windows Specific Functions

X16_DLLOpen

```
int X16_DLLOpen(void);
```

Parameters

None

Return values

Returns 0 on success. Returns an error code if any errors were detected.

Notes

Opens the DLL.

X16_DLLClose

```
void X16_DLLOpen(void);
```

Parameters

None

Return values

None

Notes

Closes the DLL.

4.2. Library Functions

4.2.1. System Functions

X16_open

```
int X16_open(void);
```

Parameters

None.

Return value

The function returns number of boards in case of successful initialization (positive), or an error code (negative).

Notes

The function initializes the driver and searches for all boards supported by the Library.

PC/104 (516) board:
Boards enumerated by the driver in order of used addresses, listed in file sx16.ini. The board number 0 is the first board found in this list. If a board was installed manually with Windows Device Manager, it will be enumerated in order of installation.

PCI (616) board:
Boards enumerated by the OS in order of used PCI slots. The board number 0 located in the slot with the least number on the PCI bus with the least number. You can run some PCI utility such as “scanpci” to check a board presence. The board IDs are: 0x1131 (vendor) and 0x7146 (device).

Mixed (516/616) boards:
Both (516/616) types of board may be used by a single application. Call X16_getboardinfo to determine the type of board.

The driver released by a call to X16_close.

X16_register

```
int X16_register(int CardId);
```

Parameters

CardId
Card identifier. Must be between 0 and number of cards minus 1.

Return values

Returns 0 on success. Returns an error code if any errors were detected.

Notes

Inits and registers the board, allocates system resources.

X16_unregister

```
void X16_unregister(int CardId);
```

Parameters

CardId
Card identifier. Must be between 0 and number of cards minus 1.

Return values

None.

Notes

Unregisters the board and frees system resources.

X16_ getboardinfo

```
void X16_getboardinfo(int CardId,BOARDINFO *info);
```

Parameters

CardId
Card identifier. Must be between 0 and number of cards minus 1.
info
Pointer to BOARDINFO structure:

```
int boardtype;          // 516/616
char *version;          // pointer to version info string
int iscapture;          // is capture support
int isoverlay;          // is overlay support
```

Return values

None.

Notes

Fills the BOARDINFO structure.

X16_close

```
void X16_close(void);
```

Parameters

None.

Return values

None.

Notes

Closes the driver.

4.2.2. Video Functions

X16_set_input

```
void X16_set_input(int CardId,int input);
```

Parameters

CardId
: Card identifier. Must be between 0 and number of cards minus 1.

input
: Video input. Must be between 0 and 3.

Return values

None.

Notes

Selects video input. Default is 0.
Sensoray 616 board (Revision A) has only one input.

X16_set_input_type

```
void X16_set_input_type(int CardId,int type);
```

Parameters

CardId
: Card identifier. Must be between 0 and number of cards minus 1.

type
: Video input type. Must be 0 for Composite Video or 1 for S-Video.

Return values

None.

Notes

Sets video input type. Default is S-Video.

X16_set_input_system

```
void X16_set_input_system(int CardId,int system);
```

Parameters

CardId
: Card identifier. Must be between 0 and number of cards minus 1.

system
: Video system. Must be PAL (1) or NTSC (2).

Return values

None.

Notes

Sets input video system NTSC or PAL. Default is 2 (NTSC).

X16_get_input_status

```
int X16_get_input_status(int CardId);
```

Parameters

CardId
: Card identifier. Must be between 0 and number of cards minus 1.

Return values

Returns the status (positive), or an error code (negative).
Status is a combination of bit flags:

Bit	Name	Function
D0	RDCAP	ready for capture (all internal loops locked); active HIGH
D1	COPRO	copy protected source detected according to macrovision version up to 7.01
D2	WIPA	white peak loop is activated; active HIGH
D3	GLIMB	gain value for active luminance channel is limited [min (bottom)]; active HIGH
D4	GLIMT	gain value for active luminance channel is limited [max (top)]; active HIGH
D5	FIDT	identification bit for detected field frequency; LOW = 50 Hz, HIGH = 60 Hz
D6	HLVLN	status bit for horizontal/vertical loop: LOW = locked, HIGH = unlocked
D7	INTL	status bit for interlace detection; LOW = non-interlaced, HIGH = interlaced

Notes

Gets status of the video input currently selected. To be sure that the input is connected to video source just check bit D0.

X16_set_output_system

```
void X16_set_output_system(int CardId,int system);
```

Parameters

CardId
Card identifier. Must be between 0 and number of cards minus 1.
system
Video system. Must be PAL (1) or NTSC (2).

Return values

None.

Notes

Sets output video system NTSC or PAL. Default is 2 (NTSC).

X16_set_bitrate

```
void X16_set_bitrate(int CardId,int rate);
```

Parameters

CardId
Card identifier. Must be between 0 and number of cards minus 1.
rate
Video bit rate in bits per second.

Return values

None.

Notes

Sets desired bit rate.

X16_set_picsize

```
void X16_set_picsize(int CardId,int size);
```

Parameters

CardId
Card identifier. Must be between 0 and number of cards minus 1.
size
Picture size index.

Return values

None.

Notes

Sets picture size and compression system:

- ***SIZE176_1*** **-** MPEG-1**,** NTSC**-176x112** PAL**-176x144**
- ***SIZE352_1*** **-** MPEG-1**,** NTSC**-352x240** PAL**-352x288**
- ***SIZE352_2*** **-** MPEG-2**,** NTSC**-352x480** PAL**-352x576**
- ***SIZE480_2*** **-** MPEG-2**,** NTSC**-480x480** PAL**-480x576**
- ***SIZE640_2*** **-** MPEG-2**,** NTSC**-640x480** PAL**-640x576**
- ***SIZE704_2*** **-** MPEG-2**,** NTSC**-704x480** PAL**-704x576**
- ***SIZE720_2*** **-** MPEG-2**,** NTSC**-720x480** PAL**-720x576**

X16_set_n

```
void X16_set_n(int CardId,int n);
```

Parameters

CardId
: Card identifier. Must be between 0 and number of cards minus 1.

n
: Number of frames in Group of Pictures.

Return values

None.

Notes

Sets number of frames in Group of Pictures. Default is 3.

X16_set_m

```
void X16_set_m(int CardId,int m);
```

Parameters

CardId
: Card identifier. Must be between 0 and number of cards minus 1.

m
: Distance between I/P frames.

Return values

None.

Notes

Sets distance between I/P frames. Default is 15.

X16_set_mux

```
void X16_set_mux(int CardId,int mux);
```

Parameters

CardId
: Card identifier. Must be between 0 and number of cards minus 1.

mux
: Multiplexing format.

Return values

None.

Notes

Multiplexing formats:

- 0: SVCD
- 1: VCD
- 2: DVD
- 3: CDDA
- 4: video ES (video only)
- 5: audio ES (audio only)
- 6: Transport Stream
- 7: 2K buffer (custom)
- 8: custom

Default is 2.

X16_set_vbr

```
void X16_set_vbr(int CardId,int vbr);
```

Parameters

CardId
: Card identifier. Must be between 0 and number of cards minus 1.

vbr
: Enables (1) or disables (0) variable bit rate compression

Return values

None.

Notes

Enables/disables variable bit rate compression. Default is 0.

X16_read

```
int X16_read(int CardId,char *buffer,int nbytes);
```

Parameters

CardId
Card identifier. Must be between 0 and number of cards minus 1.
buffer
Pointer to input video buffer.
nbytes
Number of bytes to input.

Return values

Number of bytes actually read or -1 if any errors was detected.

Notes

Attempts to read "nbytes" bytes of MPEG data from the SM2288 output FIFO. The actual amount read is returned. If no data is available “0” is returned. If this is the first read the encoder is initialized and started. The hardware transfers data by 32768 byte blocks. To avoid buffer overflow user must allocate “buffer” with size in multiple of 32K bytes.

X16_write

```
int X16_write(int CardId,char *buffer, int nbytes);
```

Parameters

CardId
Card identifier. Must be between 0 and number of cards minus 1.
buffer
Pointer to output video buffer.
nbytes
Number of bytes to output.

Return values

Number of bytes actually written or -1 if any errors was detected.

Notes

Attempts to write "nbytes" bytes of MPEG data to the SM2288 input FIFO. The actual amount written is returned. If no room in the board buffer is available “0” is returned. If this is the first write the decoder is initialized and started.
The hardware transfers data by 32768 byte blocks. To avoid buffer underflow user must allocate “buffer” with size in multiple of 32K bytes.

X16_stop

```
int X16_stop(int CardId);
```

Parameters

CardId
Card identifier. Must be between 0 and number of cards minus 1.

Return values

Returns 0 on success. Returns an error code if any errors were detected.

Notes

Stops read/write. Sends “STOP” command to SM2288.

X16_pause

```
int X16_pause(int CardId);
```

Parameters

CardId
: Card identifier. Must be between 0 and number of cards minus 1.

Return values

Returns 0 on success. Returns an error code if any errors were detected.

Notes

Pauses read/write. Sends “PAUSE” command to SM2288.

X16_resume

```
int X16_resume(int CardId);
```

Parameters

CardId
: Card identifier. Must be between 0 and number of cards minus 1.

Return values

Returns 0 on success. Returns an error code if any errors were detected.

Notes

Resume read/write after pause. Sends “RESUME” command to SM2288.

X16_set_brightness

```
void X16_set_brightness(int CardId,int brightness);
```

Parameters

CardId
: Card identifier. Must be between 0 and number of cards minus 1.

brightness
: Brightness. Must be between 0 and 255.

Return values

None.

Notes

Sets brightness on video input.

X16_set_contrast

```
void X16_set_contrast(int CardId,int contrast);
```

Parameters

CardId
: Card identifier. Must be between 0 and number of cards minus 1.

contrast
: Contrast. Must be between 0 and 255.

Return values

None.

Notes

Sets contrast on video input.

X16_set_saturation

```
void X16_set_saturation(int CardId,int saturation);
```

Parameters

CardId
: Card identifier. Must be between 0 and number of cards minus 1.

saturation
: Saturation. Must be between 0 and 255.

Return values

None.

Notes

Sets color saturation on video input.

X16_set_colorbars

```
void X16_set_colorbars (int CardId,int bars);
```

Parameters

CardId
: Card identifier. Must be between 0 and number of cards minus 1.

bars
: Bars on/off. Must be 0 or 1.

Return values

None.

Notes

Video output internal color bar generator on/off.

4.2.3. Audio Functions

X16_set_volume

```
void X16_set_volume(int CardId,int volume);
```

Parameters

CardId
Card identifier. Must be between 0 and number of cards minus 1.
volume
Volume. Must be between 0 and 63.

Return values

None.

Notes

Sets audio volume in dB (0 - maximum, 63 – minimum).

X16_set_mute

```
void X16_set_mute(int CardId,int mute);
```

Parameters

CardId
Card identifier. Must be between 0 and number of cards minus 1.
mute
Mute. Must be 0 or 1.

Return values

None.

Notes

Sets audio mute (0 – sound is on, 1 – sound is off).

X16_set_audio_pars

```
void X16_set_mute(int CardId, AUDIOPAR *par);
```

Parameters

CardId
Card identifier. Must be between 0 and number of cards minus 1.

par
Pointer to parameter table.

Return values

None.

Notes

AUDIOPAR is array of 8 chars contains Audio Control parameters:

Byte	Bit-range	Description	Default value
0	[7:0]	Audio bit rate (see details in next table)	0x0b
1	[7:4]	Audio encoding format 0: MPEG-1 Layer I 1: MPEG-1 Layer II 2: Reserved 3: Dolby Digital (AC3) 2-channel	0x10
	[3:0]	Reserved	
2	[7:6]	Stereo Mode For AC3 0: single channel 1: 2 channels others: reserved For MPEG audio 0: stereo 1: Joint stereo (intensity or ms) 2: Dual channel 3: single channel	0x00
	[5:4]	Reserved	
	[3:0]	Reserved	
3	[7:4]	Audio sampling rate 0: 44.1 kHz 1: 48 kHz 2: 32 kHz	0x00
	[3:0]	PCM Resolution 0: 16-bit 1: 18-bit 2: 20-bit 3: 24-bit	
4	[7:0]	Reserved	0x00
5	[7:0]	Audio Advanced Settings MPEG (see the MPEG spec for details) Bit [7]: CRC protection, default 1 Bit [6]: Reserved Bit [5]: Copyright, default 1 Bit [4]: Original (1) or copy (0), default 1 Bit [3]: Emphasis, default 0 AC-3 Bits [7:6]: Audio coding mode, default 2 Bits [5:1]: Bandwidth code, default 15 Bit [0] : Auto gain control, default 0	0xb0
6, 7	[7:0]	Reserved	0x00

Byte 0 value	**Bit Rate (kbps)**	
MPEG	Layer I	Layer II
1 2 3 4 5 6 7 8 9 10 11 12 13 14	32 64 96 128 160 192 224 256 288 320 352 384 416 448	32 48 56 64 80 96 112 128 160 192 224 256 320 384
AC-3		
0,1 2,3 4,5 6,7 8,9 10,11 12,13 14,15 16,17 18,19 20,21 22,23 24,25 26,27 28,29 30,31 32,33 34,35 36,37	32 40 48 56 64 80 96 112 128 160 192 224 256 320 384 448 512 576 640	

4.2.4. Capture Functions

Note: only 616 board supports the Capture Functions.

X16_prepare_capture

```
int X16_prepare_capture (int CardId, BUFFER *buffer);
```

Parameters

CardId
Card identifier. Must be between 0 and number of cards minus 1.
buffer
Pointer to a BUFFER structure. Fields:

```
unsigned char *capbuf;      //    pointer to capture (uncompressed) buffer
int capsize;                //    size of buffer in bytes
int width, height;          //    picture size in pixels
```

Return values

Returns 0 on success. Returns an error code if any errors were detected.

Notes

Prepares a buffer to capture uncompressed frames. User must specify desirable *width* and *height*. Function calculates compatible *width* and *height* (maybe different) and allocates *capbuf* with *capsize* (in bytes).

X16_get_capture_status

```
void X16_get_capture_status(int CardId);
```

Parameters

CardId
Card identifier. Must be between 0 and number of cards minus 1.

Return values

Capture status is active (1) during data transferring to the buffer.

Notes

Status set to 1 by X16_capture_start and cleared by hardware.

X16_capture_start

```
int X16_capture_start (int CardId);
```

Parameters

CardId
Card identifier. Must be between 0 and number of cards minus 1.

Return values

Returns 0 on success. Returns an error code if any errors were detected.

Notes

Starts the capture for a single frame and sets up the internal flag (*X16_get_capture_status* returns value of this flag). When the capture is done, the flag is cleared by hardware.

X16_close_capture

```
void X16_close_capture (int CardId);
```

Parameters

CardId
Card identifier. Must be between 0 and number of cards minus 1.

Return values

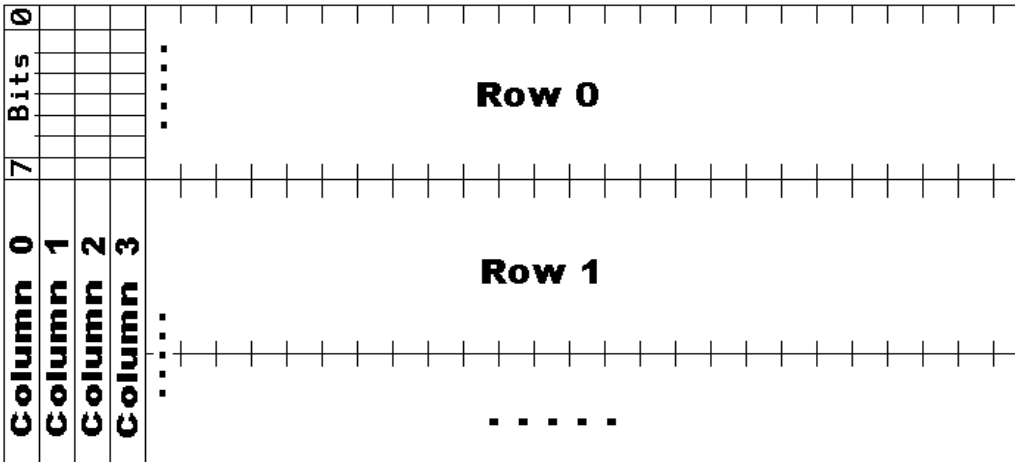
Returns 0 on success. Returns an error code if any errors were detected.

Notes

Closes and clears the caption buffer.

4.2.5. Overlay Functions

Note: not all boards support the Overlay Functions. Please your order and the hardware manual.



Overlay structure

X16_overlay_get

```
int X16_overlay_get(int CardId,int *width,int *height);
```

Parameters

CardId
: Card identifier. Must be between 0 and number of cards minus 1.

width
: Pointer to overlay image width.

height
: Pointer to overlay image height.

Return values

Returns 0 on success. Returns an error code if any errors were detected.

Notes

Fills *width* and *height* by actual overlay dimensions (if supported).

X16_overlay_setmode

int X16_overlay_setmode(int CardId,int mode);

Parameters

CardId
Card identifier. Must be between 0 and number of cards minus 1.
mode
1 - on, 0 – off.

Return values

Returns 0 on success. Returns an error code if any errors were detected.

Notes

Sets overlay on or off.

X16_overlay_setpos

int X16_overlay_setpos(int CardId,int xstart,int ystart,int xend,int yend);

Parameters

CardId
Card identifier. Must be between 0 and number of cards minus 1.
xstart, ystart
Coordinates of upper left position.
xend, yend

Return values

Returns 0 on success. Returns an error code if any errors were detected.

Notes

Sets overlay position on the screen.

X16_overlay_setcolors

int X16_overlay_setcolors(int CardId,int yfirst,int ynext,int c1,int c2);

Parameters

CardId
Card identifier. Must be between 0 and number of cards minus 1.
yfirst

luminance level (Y) for 1st pixel (0...255).

ynext
luminance level (Y) for next pixels (0...255).

c1
chrominance level (Cr) (1...255).

c2
chrominance level (Cb) (1...255).

Return values

Returns 0 on success. Returns an error code if any errors were detected.

Notes

Sets overlay colors.

Standard colors:

Color	Y	Cb	Cr
White	235	128	128
Yellow	210	16	146
Cyan	170	166	16
Green	145	54	34
Magenta	106	202	222
Red	81	90	240
Blue	41	240	110
Black	16	128	128

X16_overlay_write

```
int X16_overlay_write(int CardId,int row,int column,int data);
```

Parameters

CardId
Card identifier. Must be between 0 and number of cards minus 1.

row
Overlay row.

column
Overlay column.

data
Data to write.

Return values

Returns 0 on success. Returns an error code if any errors were detected.

Notes

Writes 8 (one byte) pixels to overlay buffer.

4.2.6. Digital I/O Functions

X16_read_gpio

```
int X16_read_gpio(int CardId);
```

Parameters

CardId
: Card identifier. Must be between 0 and number of cards minus 1.

Return values

State of GPIO

Notes

Reads state of GPIO pins: active input is low (when input is pulled down bit is set to 1).

X16_write_gpio

```
void X16_write_gpio(int CardId,int state);
```

Parameters

CardId
: Card identifier. Must be between 0 and number of cards minus 1.

state
: State of GPIO

Return values

None.

Notes

Sets state of GPIO pins: active output is low (output is pulled down when bit is set to 1).