

Sensoray Co., Inc.

311/611/711 DOS DRIVER

V0.9 12/20/99

© 1999

Table of contents

1.0	INSTALLING THE X11 DOS DEVICE DRIVER.....	2
2.0	THE DEVICE DRIVER INTERFACE.....	3
3.0	THE PARAMETER DATA STRUCTURES.....	4
3.1	MODE STRUCTURE.....	4
3.2	MODE_ADVANCED STRUCTURE.....	5
4.0	ACQUISITION AND SETUP FUNCTIONS.....	11
4.1	X11_INITSYSTEM.....	11
4.2	X11_SETMODE.....	12
4.3	X11_GETDATAPTR.....	13
4.4	X11_GETDATAVALUE.....	14
4.5	X11_SETDATAVALUE.....	15
4.6	X11_ACQUIRE.....	16
4.7	X11_STARTACQUIRE.....	17
4.8	X11_STOPACQUIRE.....	18
4.9	X11_GETSTATUS.....	19
4.10	X11_RESETSTATUS.....	20
4.11	X11_GETDSTATUS.....	21
4.12	X11_SETDSTATUS.....	22
4.13	X11_INTERRUPTON.....	23
4.14	X11_INTERRUPTMASK.....	24
4.15	X11_INTERRUPTUNMASK.....	25
4.16	X11_GETINTSTATUS.....	26
4.17	X11_INTERRUPTOFF.....	27
4.18	X11_GETRC.....	28
4.19	X11_GETIMAGESIZE.....	29
4.20	X11_MOVEIMAGE.....	30
4.21	X11_GETFRAMEADDR.....	31
4.22	X11_WRITEPORT.....	32
4.23	X11_READPORT.....	33
5.0	DISPLAY FUNCTIONS.....	34
5.1	X11_DISPOPEN.....	34
5.3	X11_DISP.....	35
5.4	X11_PUTPIXEL.....	36
5.5	X11_GETTPIXEL.....	37

1.0 Installing the x11 DOS Device Driver

The X11DOS.SYS file must be placed in a suitable directory that is accessible at boot time.

The config.sys file then must be modified to include a DEVICE=X:\PATH\X11DOS.SYS line for each Sensoray 311/611/711 frame grabber installed in the system. For the X11DOS.SYS to function an XMS driver (HIMEM.SYS) must have been installed just before the X11DOS.SYS line. Other memory managers (e.g. EMM386.EXE) can be installed, but only after X11DOS.SYS.

```
DEVICE=C:\DOS\HIMEM.SYS
DEVICE=X:\PATH\X11DOS.SYS D:0      ;First x11 card
DEVICE=X:\PATH\X11DOS.SYS D:1      ;Second x11 card
DEVICE=other memory managers
```

The Sensoray device driver line can include several optional parameters.

```
DEVICE=X:\PATH\X11DOS.SYS [F:x] [N:xxxx] [A:xxxx] [D:x]
```

Where X:\PATH\ is the path to the device driver and the options are bracketed.

A:xxxx where **xxxx** is the segment part of the base address of the Sensoray model 311/611/711. These cards require a 4KB address space. When multiple cards are installed in a system each card must have a different base address. The address must not conflict with any other adapters in the system (i.e. network, video, communication cards, etc.) By default a base address of D800:0 is assumed. **IMPORTANT:** The hex digits must be in uppercase.

F:x where **x** is the number of frame buffers to reserve. A maximum of 8 frame buffers can be specified. This value should be kept to the minimum required, as once the memory is reserved it cannot be used for anything else. The default value is 2.

N:xxxx where **xxxx** is the number of KBs to reserve for each frame. This should be roughly the product of the maximum horizontal resolution, vertical resolution, and the number of bytes per pixel used in the application. This can be further tuned by calling **X11_GETIMAGESIZE** from within the application and using the returned value. By default 2048KBs are reserved for each frame.

D:x where **x** is the card number used in the application to call the SDK functions. When multiple x11 frame grabbers are installed in a system each card must have its own device driver installed with a unique card number. Valid numbers are from 0 to 7 with 0 being the default.

Example:

```
DEVICE=C:\DOS\HIMEM.SYS
DEVICE=C:\X11DOS.SYS A:D000 D:0 N:512 F:2
DEVICE=C:\X11DOS.SYS A:D400 D:1 N:1024 F:2
```

The first has 0 for a card number, a base address of D000:0, and two frame buffers of 512KB each.

The second has 1 for a card number, a base address of D400:0, and two frame buffers of 1024KB each.

2.0 The Device Driver Interface

All the functions of the x11 DOS device driver are called via the x86 software interrupt mechanism. The interrupt number used for all the SDK functions **X11_INT**. To use the functions the file **SX11.H** must be included in the application source files.

```
#include <stdio.h>
#include <i86.h>
#include "sx11.h"

main()
{
    union REGS regs;

    //Input parameters
    regs.h.ah = X11_INITSYSTEM; //This is the function
    regs.h.al = 0;              //This is the card number for the specific frame grabber
                                //as specified on the device driver line
    regs.w.bx = 2;              //Number of frames to use
    regs.w.cx = FORMAT_PAL;

    int86(X11INT,&regs,&regs);

    if(regs.w.ax)                //Check AX for errors
    {
        printf("Error:%d\n", regs.w.ax);
        abort();
    }
}
```

All the system parameter a located in the device drivers memory space. If any of the parameters need to be modified or read (as in the *MODE* structure), either a far pointer to this data must be obtained or special functions must be used. The x11 DOS device driver provides both mechanisms. These mechanisms were designed to simplify the use of the device driver functions from a protected mode application using DPML and/or common DOS extenders.

3.0 The Parameter Data Structures

3.1 MODE Structure

```
typedef struct {  
    DWORD scale;  
    DWORD color;  
    DWORD store;  
    DWORD input;  
    MODE_ADVANCED advanced;  
} MODE;
```

The **MODE** structure contains information about the operation mode of the frame grabber. The settings that are not likely to be used frequently are hidden inside the advanced member.

Member	Description												
scale	Defines image scale. Can be one of the following:												
	<table><thead><tr><th>Value</th><th>Description</th></tr></thead><tbody><tr><td>SCALE_ADVANCED</td><td>Image scale is defined by the settings in the MODE_ADVANCED structure.</td></tr><tr><td>SCALE8</td><td>Full size image.</td></tr><tr><td>SCALE6</td><td>3/4 size image.</td></tr><tr><td>SCALE4</td><td>1/2 size image.</td></tr><tr><td>SCALE2</td><td>1/4 size image.</td></tr></tbody></table>	Value	Description	SCALE_ADVANCED	Image scale is defined by the settings in the MODE_ADVANCED structure.	SCALE8	Full size image.	SCALE6	3/4 size image.	SCALE4	1/2 size image.	SCALE2	1/4 size image.
	Value	Description											
	SCALE_ADVANCED	Image scale is defined by the settings in the MODE_ADVANCED structure.											
	SCALE8	Full size image.											
	SCALE6	3/4 size image.											
SCALE4	1/2 size image.												
SCALE2	1/4 size image.												
color	Defines output color format of the image. Can be one of the following:												
	<table><thead><tr><th>Value</th><th>Description</th></tr></thead><tbody><tr><td>COLOR_MONO</td><td>Monochrome image, 1 byte per pixel.</td></tr><tr><td>COLOR_RGB</td><td>Color image, 3 bytes per pixel.</td></tr></tbody></table>	Value	Description	COLOR_MONO	Monochrome image, 1 byte per pixel.	COLOR_RGB	Color image, 3 bytes per pixel.						
	Value	Description											
COLOR_MONO	Monochrome image, 1 byte per pixel.												
COLOR_RGB	Color image, 3 bytes per pixel.												
store	Not supported in the DOS version.												
input	Controls the input multiplexor. Can be one of the following:												
	<table><thead><tr><th>Value</th><th>Description</th></tr></thead><tbody><tr><td>MUX_0</td><td>S-Video input.</td></tr><tr><td>MUX_1</td><td>Video 1 input.</td></tr><tr><td>MUX_2</td><td>Video 2 input.</td></tr></tbody></table>	Value	Description	MUX_0	S-Video input.	MUX_1	Video 1 input.	MUX_2	Video 2 input.				
	Value	Description											
	MUX_0	S-Video input.											
MUX_1	Video 1 input.												
MUX_2	Video 2 input.												
advanced	MODE_ADVANCED structure. Defines the advanced mode settings.												

3.2 **MODE_ADVANCED** Structure

```
typedef struct {  
    WORD interlace;  
    WORD xTotal;  
    WORD xActive;  
    WORD xDelay;  
    WORD yFactor;  
    WORD yActive;  
    WORD yDelay;  
    WORD FORMAT;  
    WORD BRIGHT;  
    WORD CONTRAST;  
    WORD SAT_U;  
    WORD SAT_V;  
    WORD HUE;  
    WORD LNOTCH;  
    WORD LDEC;  
    WORD DEC_RAT;  
    WORD PEAK;  
    WORD CAGC;  
    WORD CKILL;  
    WORD HFILT;  
    WORD RANGE;  
    WORD CORE;  
    WORD YCOMB;  
    WORD CCOMB;  
    WORD ADELAY;  
    WORD BDELAY;  
    WORD SLEEP;  
    WORD CRUSH;  
    WORD VFILT;  
    WORD COLOR_BARS;  
    WORD GAMMA;  
    WORD PKTP;  
    WORD bimodal;  
    DWORD colorkey;  
    WORD buffertype;  
    DWORD reserved1;  
    DWORD reserved2;  
    DWORD reserved3;  
    DWORD reserved4;  
} MODE_ADVANCED;
```

The **MODE_ADVANCED** structure contains information about the advanced operation mode of the frame grabber.

Member	Description																
interlace	<p>Defines input image format. Can be one of the following:</p> <table border="0"> <thead> <tr> <th style="text-align: left;">Value</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td>IMG_INTERLACED</td> <td>Interlaced input image.</td> </tr> <tr> <td>IMG_NONINTERLACED</td> <td>Noninterlaced input image.</td> </tr> </tbody> </table>	Value	Description	IMG_INTERLACED	Interlaced input image.	IMG_NONINTERLACED	Noninterlaced input image.										
Value	Description																
IMG_INTERLACED	Interlaced input image.																
IMG_NONINTERLACED	Noninterlaced input image.																
xTotal	Total number of output horizontal pixels (including horizontal blanking). Should be between 100 and 910 (NTSC), or 1135 (PAL). This is the number of pixels generated by the frame grabber internally.																
xActive	Number of active output horizontal pixels. Should be between 80 and 900 (NTSC), or 1000 (PAL). This is the number of pixels in the output image.																
xDelay	The horizontal offset of the start of the active area relative to horizontal sync, pixels. The following condition should always be met: $xDelay + xActive \leq xTotal$.																
yFactor	Vertical scaling factor. The actual number of lines generated by the video source (525 for NTSC, 625 for PAL) is divided by yFactor to get the number of lines in the output image. Should be between 1.0 and 8.0 and then multiplied by 4096.																
yActive	Number of active output lines (before vertical scaling is applied). Should be between 60 and 525 (NTSC), or 625 (PAL). For example, to grab the whole NTSC image scaled down by the factor of 2, set yActive to 525, yFactor to 2.0 (8192). To grab the upper half of the NTSC image scaled down by the factor of 2, set yActive to 262, yFactor to 2.0 (8192).																
yDelay	The vertical offset of the start of the active area relative to vertical sync, lines (before vertical scaling is applied). The following condition should always be met: $yDelay + yActive \leq (\text{total lines, 525 or 625})$.																
FORMAT	<p>Input signal format. Can be one of the following:</p> <table border="0"> <thead> <tr> <th style="text-align: left;">Value</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td>FORMAT_NTSC</td> <td>NTSC input signal.</td> </tr> <tr> <td>FORMAT_NTSCJ</td> <td>NTSC (Japan) input signal.</td> </tr> <tr> <td>FORMAT_PAL</td> <td>PAL input signal.</td> </tr> <tr> <td>FORMAT_PALM</td> <td>PAL(M) input signal.</td> </tr> <tr> <td>FORMAT_PALN</td> <td>PAL(N) input signal.</td> </tr> <tr> <td>FORMAT_PALNC</td> <td>PAL(N-combination) input signal.</td> </tr> <tr> <td>FORMAT_SECAM</td> <td>SECAM input signal.</td> </tr> </tbody> </table>	Value	Description	FORMAT_NTSC	NTSC input signal.	FORMAT_NTSCJ	NTSC (Japan) input signal.	FORMAT_PAL	PAL input signal.	FORMAT_PALM	PAL(M) input signal.	FORMAT_PALN	PAL(N) input signal.	FORMAT_PALNC	PAL(N-combination) input signal.	FORMAT_SECAM	SECAM input signal.
Value	Description																
FORMAT_NTSC	NTSC input signal.																
FORMAT_NTSCJ	NTSC (Japan) input signal.																
FORMAT_PAL	PAL input signal.																
FORMAT_PALM	PAL(M) input signal.																
FORMAT_PALN	PAL(N) input signal.																
FORMAT_PALNC	PAL(N-combination) input signal.																
FORMAT_SECAM	SECAM input signal.																
BRIGHT	Controls the brightness (luminance) of the output signal. Takes the values between 0 and 0xFF which are treated as a signed offset, from -128 (0x80) to +127 (0x7F). The resolution of brightness change is one LSB (0.4% of the full range).																
CONTRAST	This 9-bit value is multiplied by the luminance value to provide contrast (gain) adjustment. Takes values from 0 to 0xFF (237%), with 0x0D8 corresponding to 100%.																

SAT_U	A 9-bit value used to add a gain adjustment to the U component of the video signal. By adjusting U and V color components by the same incremental value, the saturation is adjusted. Takes values between 0 and 0x1FF (201%), with 0x0FE corresponding to 100%.						
SAT_V	A 9-bit value used to add a gain adjustment to the V component of the video signal. By adjusting U and V color components by the same incremental value, the saturation is adjusted. Takes values between 0 and 0x1FF (284%), with 0x0B4 corresponding to 100%.						
HUE	Controls the hue by adjusting the demodulating subcarrier phase. Takes values between 0 and 0xFF, which are treated as a signed offset with 0x80 corresponding to -90 degrees, and 0x7F corresponding to +89 degrees.						
LNOTCH	Controls the internal luminance notch filter which attenuates the subcarrier in the output signal, removing the "checkboard" pattern in the output image, in case a composite input is used. Can be one of the following: <table border="0" style="margin-left: 20px;"> <thead> <tr> <th style="text-align: left;"><u>Value</u></th> <th style="text-align: left;"><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>LNOTCH_OFF</td> <td>Filter disabled.</td> </tr> <tr> <td>LNOTCH_ON</td> <td>Filter enabled.</td> </tr> </tbody> </table>	<u>Value</u>	<u>Description</u>	LNOTCH_OFF	Filter disabled.	LNOTCH_ON	Filter enabled.
<u>Value</u>	<u>Description</u>						
LNOTCH_OFF	Filter disabled.						
LNOTCH_ON	Filter enabled.						
LDEC	Controls the luminance decimation filter used to reduce the high-frequency components of the luma signal. Useful when scaling down to lower resolutions. See HFILT for details. Can be one of the following: <table border="0" style="margin-left: 20px;"> <thead> <tr> <th style="text-align: left;"><u>Value</u></th> <th style="text-align: left;"><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>LDEC_OFF</td> <td>Filter disabled.</td> </tr> <tr> <td>LDEC_ON</td> <td>Filter enabled.</td> </tr> </tbody> </table>	<u>Value</u>	<u>Description</u>	LDEC_OFF	Filter disabled.	LDEC_ON	Filter enabled.
<u>Value</u>	<u>Description</u>						
LDEC_OFF	Filter disabled.						
LDEC_ON	Filter enabled.						
DEC_RAT	A 6-bit value corresponding to the number of fields or frames dropped out of 60 (NTSC) or 50 (PAL/SECAM). A value of 0 disables decimation.						
PEAK	Determines whether the normal or the peaking luma low pass filters are implemented via the HFILT. Can be one of the following: <table border="0" style="margin-left: 20px;"> <thead> <tr> <th style="text-align: left;"><u>Value</u></th> <th style="text-align: left;"><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>PEAK_OFF</td> <td>Normal low pass filters.</td> </tr> <tr> <td>PEAK_ON</td> <td>Peaking low pass filters.</td> </tr> </tbody> </table>	<u>Value</u>	<u>Description</u>	PEAK_OFF	Normal low pass filters.	PEAK_ON	Peaking low pass filters.
<u>Value</u>	<u>Description</u>						
PEAK_OFF	Normal low pass filters.						
PEAK_ON	Peaking low pass filters.						
CAGC	Controls the chroma AGC function. When enabled, will compensate for nonstandard chroma levels by multiplying the incoming chroma signal by a value in the range of 0.5 to 2.0. Can be one of the following: <table border="0" style="margin-left: 20px;"> <thead> <tr> <th style="text-align: left;"><u>Value</u></th> <th style="text-align: left;"><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>CAGC_OFF</td> <td>Chroma AGC off.</td> </tr> <tr> <td>CAGC_ON</td> <td>Chroma AGC on.</td> </tr> </tbody> </table>	<u>Value</u>	<u>Description</u>	CAGC_OFF	Chroma AGC off.	CAGC_ON	Chroma AGC on.
<u>Value</u>	<u>Description</u>						
CAGC_OFF	Chroma AGC off.						
CAGC_ON	Chroma AGC on.						
CKILL	Controls the low color detector and removal circuitry. Can be one of the following: <table border="0" style="margin-left: 20px;"> <thead> <tr> <th style="text-align: left;"><u>Value</u></th> <th style="text-align: left;"><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>CKILL_OFF</td> <td>Low color detection and removal disabled.</td> </tr> <tr> <td>CKILL_ON</td> <td>Low color detection and removal enabled.</td> </tr> </tbody> </table>	<u>Value</u>	<u>Description</u>	CKILL_OFF	Low color detection and removal disabled.	CKILL_ON	Low color detection and removal enabled.
<u>Value</u>	<u>Description</u>						
CKILL_OFF	Low color detection and removal disabled.						
CKILL_ON	Low color detection and removal enabled.						

HFILT	<p>Controls the degree of horizontal low-pass filtering provided LDEC is set to LDEC_ON. Can be one of the following:</p> <table border="0"> <thead> <tr> <th style="text-align: left;"><u>Value</u></th> <th style="text-align: left;"><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>HFILT_AUTO</td> <td>The filter is selected automatically depending on the scale setting. When horizontal scaling is between full and half resolution, no filtering is selected. When scaling between one-half and one-quarter resolution, the CIF filter is used. When scaling between one-quarter and one-eighth resolution, the QCIF filter is used. When scaling below one-eighth resolution, the ICON filter is used.</td> </tr> <tr> <td>HFILT_CIF</td> <td>CIF filter.</td> </tr> <tr> <td>HFILT_QCIF</td> <td>QCIF filter.</td> </tr> <tr> <td>HFILT_ICON</td> <td>ICON filter.</td> </tr> </tbody> </table>	<u>Value</u>	<u>Description</u>	HFILT_AUTO	The filter is selected automatically depending on the scale setting. When horizontal scaling is between full and half resolution, no filtering is selected. When scaling between one-half and one-quarter resolution, the CIF filter is used. When scaling between one-quarter and one-eighth resolution, the QCIF filter is used. When scaling below one-eighth resolution, the ICON filter is used.	HFILT_CIF	CIF filter.	HFILT_QCIF	QCIF filter.	HFILT_ICON	ICON filter.
<u>Value</u>	<u>Description</u>										
HFILT_AUTO	The filter is selected automatically depending on the scale setting. When horizontal scaling is between full and half resolution, no filtering is selected. When scaling between one-half and one-quarter resolution, the CIF filter is used. When scaling between one-quarter and one-eighth resolution, the QCIF filter is used. When scaling below one-eighth resolution, the ICON filter is used.										
HFILT_CIF	CIF filter.										
HFILT_QCIF	QCIF filter.										
HFILT_ICON	ICON filter.										
RANGE	<p>Determines the range of the luminance output. Can be one of the following:</p> <table border="0"> <thead> <tr> <th style="text-align: left;"><u>Value</u></th> <th style="text-align: left;"><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>RANGE_NORM</td> <td>Normal operation (luma range 16-253).</td> </tr> <tr> <td>RANGE_FULL</td> <td>Full range operation (luma range 0-255).</td> </tr> </tbody> </table>	<u>Value</u>	<u>Description</u>	RANGE_NORM	Normal operation (luma range 16-253).	RANGE_FULL	Full range operation (luma range 0-255).				
<u>Value</u>	<u>Description</u>										
RANGE_NORM	Normal operation (luma range 16-253).										
RANGE_FULL	Full range operation (luma range 0-255).										
CORE	<p>Controls the coring value. When coring is enabled, luminance levels below a certain value are truncated to 0. Can be one of the following:</p> <table border="0"> <thead> <tr> <th style="text-align: left;"><u>Value</u></th> <th style="text-align: left;"><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>CORE_OFF</td> <td>Coring disabled.</td> </tr> <tr> <td>CORE_8</td> <td>Coring threshold is 8.</td> </tr> <tr> <td>CORE_16</td> <td>Coring threshold is 16.</td> </tr> <tr> <td>CORE_24</td> <td>Coring threshold is 24.</td> </tr> </tbody> </table>	<u>Value</u>	<u>Description</u>	CORE_OFF	Coring disabled.	CORE_8	Coring threshold is 8.	CORE_16	Coring threshold is 16.	CORE_24	Coring threshold is 24.
<u>Value</u>	<u>Description</u>										
CORE_OFF	Coring disabled.										
CORE_8	Coring threshold is 8.										
CORE_16	Coring threshold is 16.										
CORE_24	Coring threshold is 24.										
YCOMB	<p>Controls the luminance comb filtering. Can be one of the following:</p> <table border="0"> <thead> <tr> <th style="text-align: left;"><u>Value</u></th> <th style="text-align: left;"><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>YCOMB_OFF</td> <td>Vertical low-pass filtering and vertical interpolation.</td> </tr> <tr> <td>YCOMB_ON</td> <td>Vertical low-pass filtering only. The number of filter taps is determined by VFILT setting.</td> </tr> </tbody> </table>	<u>Value</u>	<u>Description</u>	YCOMB_OFF	Vertical low-pass filtering and vertical interpolation.	YCOMB_ON	Vertical low-pass filtering only. The number of filter taps is determined by VFILT setting.				
<u>Value</u>	<u>Description</u>										
YCOMB_OFF	Vertical low-pass filtering and vertical interpolation.										
YCOMB_ON	Vertical low-pass filtering only. The number of filter taps is determined by VFILT setting.										
CCOMB	<p>Controls the chrominance comb filtering. Can be one of the following:</p> <table border="0"> <thead> <tr> <th style="text-align: left;"><u>Value</u></th> <th style="text-align: left;"><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>CCOMB_OFF</td> <td>Chroma filter disabled.</td> </tr> <tr> <td>CCOMB_ON</td> <td>Chroma filter enabled.</td> </tr> </tbody> </table>	<u>Value</u>	<u>Description</u>	CCOMB_OFF	Chroma filter disabled.	CCOMB_ON	Chroma filter enabled.				
<u>Value</u>	<u>Description</u>										
CCOMB_OFF	Chroma filter disabled.										
CCOMB_ON	Chroma filter enabled.										
ADELAY	<p>Back-porch sampling delay. The default values are 0x68 (NTSC), and 0x7F (PAL/SECAM).</p>										
BDELAY	<p>Subcarrier sampling delay. The default values are 0x5D (NTSC), and 0x73 (PAL/SECAM).</p>										

SLEEP Controls sleep mode of luma and chroma A/D's. Can take the following values:

<u>Value</u>	<u>Description</u>
SLEEP_OFF	Both A/D's operating.
Y_SLEEP	Luma A/D in sleep mode.
C_SLEEP	Chroma A/D in sleep mode. Y_SLEEP and C_SLEEP can be ORed, to disable both A/D's.

CRUSH Controls the AGC mode. Can be one of the following:

<u>Value</u>	<u>Description</u>
CRUSH_OFF	Nonadaptive AGC.
CRUSH_ON	Adaptive AGC. Overflows in A/D's result in the input voltage range increase.

VFILT Controls the number of taps in the vertical scaling filter. Can be one of the following:

<u>Value</u>	<u>Description</u>
If YCOMB is set to YCOMB_ON :	
VFILT_0	2-tap filter.
VFILT_1	3-tap filter. Only available if scaling to less than 385 horizontal active pixels.
VFILT_2	4-tap filter. Only available if scaling to less than 193 horizontal active pixels.
VFILT_3	5-tap filter. Only available if scaling to less than 193 horizontal active pixels.
If YCOMB is set to YCOMB_OFF :	
VFILT_0	2-tap interpolation only.
VFILT_1	2-tap filter and 2-tap interpolation. Only available if scaling to less than 385 horizontal active pixels.
VFILT_2	3-tap filter and 2-tap interpolation. Only available if scaling to less than 193 horizontal active pixels.
VFILT_3	4-tap filter and 2-tap interpolation. Only available if scaling to less than 193 horizontal active pixels.

COLOR_BARS Controls a test color bar pattern. Can be one of the following:

<u>Value</u>	<u>Description</u>
COLORBARS_OFF	Color bars off.
COLORBARS_ON	Color bars on.

GAMMA Controls gamma correction removal. Can be one of the following:

<u>Value</u>	<u>Description</u>
GAMMA_REMOVE_ON	Gamma correction removal on.
GAMMA_REMOVE_OFF	Gamma correction removal off.

PKTP FIFO trigger point. Can be one of the following:

<u>Value</u>	<u>Description</u>
PKTP4	4 DWORDS.
PKTP8	8 DWORDS.
PKTP16	16 DWORDS.
PKTP32	32 DWORDS.

4.0 ACQUISITION AND SETUP FUNCTIONS

4.1 X11_INITSYSTEM

The **X11_INITSYSTEM** function performs the system initialization. It sets the number of image frames to be used and initializes internal data structures.

Entry	Description	Value
AH	Function number	X11_INITSYSTEM
AL	Device driver/board number	First board in system 0, second 1, etc.
BX	Number of frames	From 1 to the maximum set on the device line in the config.sys file.
CX	Default video system	FORMAT_NTSC , FORMAT_PAL , or FORMAT_SECAM .
Exit	Description	Value
AX	Completion status	Zero if successful. Error code if failed.

Note: This function must be called before any other X11 function.

Example:

```
#include <stdio.h>
#include <i86.h>
#include "sx11.h"

main()
{
    union REGS regs;

    regs.h.ah = X11_INITSYSTEM;
    regs.h.al = 0;
    regs.w.bx = 2;
    regs.w.cx = PALSECAM;

    int86(X11INT,&regs,&regs);

    if(regs.w.ax)
    {
        printf("Error:%d\n", regs.w.ax);
        abort();
    }
}
```

4.2 X11_SETMODE

The **X11_SETMODE** function sets the required frame grabber mode.

Entry	Description	Value
AH	Function number	X11_INITSYSTEM
AL	Device driver/board number	First board in system 0, second 1, etc.
Exit	Description	Value
AX	Completion status	Zero if successful. Error code if failed.

Note: **MODE** members can be modified as a result of the call to **X11_SETMODE**. For example, if one of the predefined scale settings is used (**SCALE8**, etc.), the members of the advanced portion of **MODE** are set accordingly. A pointer to the global **MODE** structure must be obtained to view or modify any of the **MODE** members. See **X11_GETMODEPTR**

Example:

```
-  
-  
MODE _far *pmode;  
-  
-  
regs.h.ah = X11_GETDATAPTR;  
regs.h.al = 0;  
regs.w.bx = MODE_PTR;  
int86(X11_INT,&regs,&regs);  
pmode = MK_FP(regs.w.dx,regs.w.cx);  
-  
-  
pmode->input = 1;  
pmode->scale = SCALE6;  
pmode->color = COLOR_RGB16;  
  
regs.h.ah = X11_SETMODE;  
regs.h.al = 0;  
int86(X11_INT,&regs,&regs);  
-  
-
```

4.3 X11_GETDATAPTR

The **X11_GETDATAPTR** function obtains a far pointer to the requested data structure or variable. A far pointer to the data permits the modification or inspection of the data or in the case of a structure of any of its members.

Entry	Description	Value
AH	Function number	X11_GETDATAPTR
AL	Device driver/board number	First board in system 0, second 1, etc.
BX	Data structure pointer requested	MODE_PTR for a pointer to the mode Structure. INTFLAG_PTR for a pointer to the interrupt flag. RCBUF_PTR for a pointer to the the buffer holding image data after a call to X11_GETRC .
Exit	Description	Value
AX	Completion status	Zero if successful. Error code if failed.
DX:CX	far pointer to the data data structure	DX has the segment and CX the offset in standard x86 real mode format. The C macro MK_FP can be used to make A far pointer.

Example:

```
-  
-  
MODE_far *pmode;  
-  
-  
regs.h.ah = X11_GETDATAPTR;  
regs.h.al = 0;  
regs.w.bx = MODE_PTR;  
int86(X11_INT,&regs,&regs);  
pmode = MK_FP(regs.w.dx,regs.w.cx);  
-  
-  
pmode->input = 1;  
pmode->scale = SCALE6;  
pmode->color = COLOR_RGB16;  
  
printf("%d\n",pmode->advanced.xTotal);
```

4.4 X11_GETDATAVALUE

The **X11_GETDATAVALUE** function obtains the value of global data. This is an alternate method for reading mode parameters and other data. This may be a simpler way of modifying the mode structure members in protected mode.

Entry	Description	Value
AH	Function number	X11_GETDATAVALUE
AL	Device driver/board number	First board in system 0, second 1, etc.
BX	Address of data to read	See below.

Exit	Description	Value
AX	Completion status	Zero if successful. Otherwise failed.
DX:CX	double word value	

<u>MODE members</u>	<u>MODE_ADVANCED members</u>	<u>Other data</u>	<u>Description</u>
MODE_SCALE	AMODE_INTERLACE	SCALE_XSIZE	Image width in pixels.
MODE_COLOR	AMODE_XTOTAL	SCALE_YSIZE	Image height in pixels.
MODE_INPUT	AMODE_XACTIVE	ISYS_IBPP	Bytes per pixel.
	AMODE_XDELAY	ISYS_SLOT	Slot number for card.
	AMODE_YFACTOR	ISYS_BASESEG	x11 base address.
	AMODE_YACTIVE	ISYS_MAXFRAMES	Max. number of frames.
	AMODE_YDELAY	ISYS_FRAME_SIZE	Size of each frame.
	AMODE_FORMAT	ISYS_FRAMECOUNT	Actual number of frames used.
	AMODE_BRIGHT	INTD_HWINT	Hardware IRQ number.
	AMODE_CONTRAST	INTD_INTFLAG	Interrupt flag.
	AMODE_SAT_U		
	AMODE_SAT_V		
	AMODE_HUE		
	AMODE_LNOTCH		
	AMODE_LDEC		
	AMODE_DEC_RAT		
	AMODE_PEAK		
	AMODE_CAGC		
	AMODE_CKILL		
	AMODE_HFILT		
	AMODE_RANGE		
	AMODE_CORE		
	AMODE_YCOMB		
	AMODE_CCOMB		
	AMODE_ADELAY		
	AMODE_BDELAY		
	AMODE_SLEEP		
	AMODE_CRUSH		
	AMODE_VFILT		
	AMODE_COLOR_BARS		
	AMODE_GAMMA		
	AMODE_PKTP		
	AMODE_BIMODAL		

4.5 X11_SETDATAVALUE

The **X11_SETDATAVALUE** function set the value of global data. This is an alternate method for setting mode parameters and other data. This may be a simpler way of modifying the **MODE** structure members in protected mode.

Entry	Description	Value
AH	Function number	X11_SETDATAVALUE
AL	Device driver/board number	First board in system 0, second 1, etc.
BX	Address of data to set	See below.
DX:CX	Double word value	
Exit	Description	Value
AX	Completion status	Zero if successful. Otherwise failed.

MODE members	MODE_ADVANCED members	Other data	Description
MODE_SCALE	AMODE_INTERLACE	INTD_INTFLAG	Interrupt flag.
MODE_COLOR	AMODE_XTOTAL		
MODE_INPUT	AMODE_XACTIVE		
	AMODE_XDELAY		
	AMODE_YFACTOR		
	AMODE_YACTIVE		
	AMODE_YDELAY		
	AMODE_FORMAT		
	AMODE_BRIGHT		
	AMODE_CONTRAST		
	AMODE_SAT_U		
	AMODE_SAT_V		
	AMODE_HUE		
	AMODE_LNOTCH		
	AMODE_LDEC		
	AMODE_DEC_RAT		
	AMODE_PEAK		
	AMODE_CAGC		
	AMODE_CKILL		
	AMODE_HFILT		
	AMODE_RANGE		
	AMODE_CORE		
	AMODE_YCOMB		
	AMODE_CCOMB		
	AMODE_ADELAY		
	AMODE_BDELAY		
	AMODE_SLEEP		
	AMODE_CRUSH		
	AMODE_VFILT		
	AMODE_COLOR_BARS		
	AMODE_GAMMA		
	AMODE_PKTP		
	AMODE_BIMODAL		

4.6 X11_ACQUIRE

The **X11_ACQUIRE** function grabs the number of frames corresponding to that of the image buffer.

<u>Entry</u>	<u>Description</u>	<u>Value</u>
AH	Function number	X11_ACQUIRE
AL	Device driver/board number	First board in system 0, second 1, etc.
DX	Timeout	Timeout in 50ms ticks. e.g DX=20 would give a timeout of about 1 second.
<u>Exit</u>	<u>Description</u>	<u>Value</u>
AX	Completion status	Zero if successful. Error code if failed.

The function returns after the acquisition is complete, or timeout expires. The function also returns the status with the value of the status word corresponding to the end of the acquisition of the last frame. Status bits are not reset automatically between the acquisition of the individual frames. See **X11_GETSTATUS** for the description of the constants used to select individual status bits.

4.7 X11_STARTACQUIRE

The **X11_STARTACQUIRE** function starts the acquisition of the number of frames corresponding to that of the image buffer, and returns immediately.

Entry	Description	Value
AH	Function number	X11_STARTACQUIRE
AL	Device driver/board number	First board in system 0, second 1, etc.
DX	Acquisition mode	AMODE_SINGLE Image buffer is filled once. AMODE_CONT Image buffer is filled continuously, until the acquisition is stopped.

Exit	Description	Value
AX	Completion status	Zero if successful. Error code if failed.

The function returns immediately after the acquisition is started. The application determines if the acquisition is complete by polling status bits, or through the use of interrupts. If continuous mode is selected, the first frame of the image buffer starts being overwritten as soon as the last frame gets filled. The application determines the completion of each individual frame by polling (and resetting) the **STATUS_READY** bit, or through the use of the interrupts. If **AMODE_SINGLE** is selected, the **STATUS_READY_ALL** bit is set upon the completion of the last frame of the image buffer. If **AMODE_CONT** is selected, the **STATUS_READY_ALL** bit can not be polled reliably, because it is being reset at the start of the first frame acquisition. There is no interrupt associated with the **STATUS_READY_ALL** bit.

4.8 X11_STOPACQUIRE

The X11_StopAcquire function stops the image acquisition. It has to be used only if acquisition was started by calling X11_StartAcquire.

Entry	Description	Value
AH	Function number	X11_STOPACQUIRE
AL	Device driver/board number	First board in system 0, second 1, etc.

Exit	Description	Value
AX	Completion status	Zero if successful. Error code if failed.

4.9 X11_GETSTATUS

The **X11_GETSTATUS** function retrieves the value of the frame grabber status word.

Entry	Description	Value
AH	Function number	X11_GETSTATUS
AL	Device driver/board number	First board in system 0, second 1, etc.

Exit	Description	Value
AX	Completion status	Zero if successful. Error code if failed.
CX:BX	32-bit status value	See below for return values.

The individual bits of the status word have the following meanings:

Value	Description
STATUS_READY	Frame acquisition complete. Has to be reset by the application. See X11_ResetStatus.
STATUS_READY_ALL	Image buffer acquisition complete. Reset automatically at the start of the first frame acquisition.
STATUS_VIDEO	Video status changed at the input (e.g. present to absent).
STATUS_HLOCK	Horizontal lock condition changed at the input.
STATUS_OFLOW	Overflow detected.
STATUS_HSYNC	Start of new line.
STATUS_VSYNC	Start of new field.
STATUS_FMT	Video format change detected (e.g. NTSC to PAL).
STATUS_ERROR	Transfer error occurred. This is a combination of bits.

4.10 X11_RESETSTATUS

The **X11_RESETSTATUS** function resets individual bits of the frame grabber status register.

Entry	Description	Value
AH	Function number	X11_RESETSTATUS
AL	Device driver/board number	First board in system 0, second 1, etc.
DI:DX	Mask value	A value of 1 resets the corresponding bit of the status register.

Exit	Description	Value
AX	Completion status	Zero if successful. Error code if failed.

Note: For the meanings of the status word individual bits see **X11_GETSTATUS**.

4.11 X11_GETDSTATUS

The **X11_GETDSTATUS** function retrieves additional status information.

Entry	Description	Value
AH	Function number	X11_GETDSTATUS
AL	Device driver/board number	First board in system 0, second 1, etc.

Exit	Description	Value
AX	Completion status	Zero if successful. Error code if failed.
CX:BX	32-bit status value	See below for return values.

The individual bits of the status word have the following meanings:

Value	Description
DSTATUS_PRES	Video present. Reset to 0 when input sync is not detected in 31 consecutive line periods.
DSTATUS_HLOCK	Device in horizontal lock.
DSTATUS_FIELD	Reflects whether an odd or even field is being captured (0 - odd field).
DSTATUS_NUML	Number of lines found in input video signal (0 - 525, 1 - 625).
DSTATUS_CSEL	Identifies which crystal is selected.
DSTATUS_LOF	Luma ADC overflow. Set to 0 at the reset, set to 1 if an overflow occurs. Has to be written to to be reset.
DSTATUS_COF	Chroma ADC overflow. Set to 0 at the reset, set to 1 if an overflow occurs. Has to be written to to be reset.

Note: Use **X11_SETDSTATUS** to reset status bits.

4.12 X11_SETDSTATUS

The **X11_SETDSTATUS** function modifies individual bits of the frame grabber **DSTATUS** register.

<u>Entry</u>	<u>Description</u>	<u>Value</u>
AH	Function number	X11_SETDSTATUS
AL	Device driver/board number	First board in system 0, second 1, etc.
CX:BX	Status value	Status value to be written.
DI:DX	Mask value	A value of 1 resets the corresponding bit of the DSTATUS register.

<u>Exit</u>	<u>Description</u>	<u>Value</u>
AX	Completion status	Zero if successful. Error code if failed.

For the meanings of the status word individual bits see **X11_GETDSTATUS**.

4.13 X11_INTERRUPTON

The **X11_INTERRUPTON** function enables the interrupts for the frame grabber.

Entry	Description	Value
AH	Function number	X11_INTERRUPTON
AL	Device driver/board number	First board in system 0, second 1, etc.
DX:BX	User interrupt handler address	DX:BX are the segment:offset pair of the handler. If DX is zero the interrupt is enabled but no user handler is required.

Exit	Description	Value
AX	Completion status	Zero if successful. Error code if failed.

Note: This function does not unmask or mask the interrupt enable register but simply sets up the system to accept interrupts. See **X11_INTERRUPTMASK** and **X11_INTERRUPTUNMASK**.

Note: When an interrupt occurs the **INTFLAG** flag is set (see **X11_GETDATAPTR**), but once set it must be cleared manually. When **X11_INTERRUPTON** is called with a DX value of zero, **INTFLAG** can be used to provide minimal real-time performance.

4.14 X11_INTERRUPTMASK

The **X11_INTERRUPTMASK** function masks the interrupts for the frame grabber.

Entry	Description	Value
AH	Function number	X11_INTERRUPTMASK
AL	Device driver/board number	First board in system 0, second 1, etc.
CX:BX	Interrupt source(s) to mask value	STATUS_READY Frame acquisition complete. STATUS_VIDEO Video status changed at the input (e.g. present to absent). STATUS_HLOCK Horizontal lock condition changed at the input. STATUS_OFLOW Overflow detected. STATUS_HSYNC Start of new line. STATUS_VSYNC Start of new field. STATUS_FMT Video format change Detected (e.g. NTSC to PAL). STATUS_ERROR Transfer error occurred. This is a combination of bits.
Exit	Description	Value
AX	Completion status	Zero if successful. Error code if failed.
CX:BX	Current mask	See above mask values.

Note: The mask values can be ORed to mask multiple interrupt sources.

4.15 X11_INTERRUPTUNMASK

The **X11_INTERRUPTUNMASK** function unmask the interrupts for the frame grabber.

Entry	Description	Value
AH	Function number	X11_INTERRUPTUNMASK
AL	Device driver/board number	First board in system 0, second 1, etc.
CX:BX	Interrupt source(s) to unmask value	STATUS_READY Frame acquisition complete. STATUS_VIDEO Video status changed at the input (e.g. present to absent). STATUS_HLOCK Horizontal lock condition changed at the input. STATUS_OFLOW Overflow detected. STATUS_HSYNC Start of new line. STATUS_VSYNC Start of new field. STATUS_FMT Video format change Detected (e.g. NTSC to PAL). STATUS_ERROR Transfer error occurred. This is a combination of bits.
Exit	Description	Value
AX	Completion status	Zero if successful. Error code if failed.
CX:BX	Current mask	See above mask values.

Note: The mask values can be ORed to unmask multiple interrupt sources.

4.16 X11_GETINTSTATUS

The **X11_GETINTSTATUS** function returns the interrupt status of the last interrupt.

Entry	Description	Value
AH	Function number	X11_GETINTSTATUS
AL	Device driver/board number	First board in system 0, second 1, etc.
Exit	Description	Value
AX	Completion status	Always returns zero.
CX:BX	Interrupt status	STATUS_READY Frame acquisition complete. STATUS_VIDEO Video status changed at the input (e.g. present to absent). STATUS_HLOCK Horizontal lock condition changed at the input. STATUS_OFLOW Overflow detected. STATUS_HSYNC Start of new line. STATUS_VSYNC Start of new field. STATUS_FMT Video format change Detected (e.g. NTSC to PAL). STATUS_ERROR Transfer error occurred. This is a combination of bits.

4.17 X11_INTERRUPTOFF

The **X11_INTERRUPTOFF** function disables the interrupts for the frame grabber.

Entry	Description	Value
AH	Function number	X11_INTERRUPTOFF
AL	Device driver/board number	First board in system 0, second 1, etc.

Exit	Description	Value
AX	Completion status	Always returns zero.

Note: This function must be called before terminating your application if interrupts have been enabled, see **X11_INTERRUPTON**.

4.18 X11_GETRC

The **X11_GETRC** function retrieves the data corresponding to one row or column of an image buffer frame into the **RCBUF** buffer (see **X11_GETDATAPTR**).

Entry	Description	Value
AH	Function number	X11_GETRC
AL	Device driver/board number	First board in system 0, second 1, etc.
BX	Frame number	
CX	X11_GETRC function	RMODE_ROW Retrieves row data RMODE_COL Retrieves column data RMODE_RLEN Retrieves row length in bytes only. RMODE_CLEN Retrieves column length in bytes only.
Exit	Description	Value
AX	Completion status	Zero if successful. Error code if failed.
BX	Number of bytes transferred.	
CX	Number of bytes per pixel.	

Note: Use **X11_GETDATAPTR** with **RCBUF_PTR** to obtain a far pointer to **RCBUF**.

4.19 X11_GETIMAGESIZE

The **X11_GETIMAGESIZE** function returns the size of an image with the current mode parameters.

Entry	Description	Value
AH	Function number	X11_GETIMAGESIZE
AL	Device driver/board number	First board in system 0, second 1, etc.
Exit	Description	Value
AX	Completion status	Zero if successful. Error code if failed.
CX:BX	Image size	.

4.20 X11_MOVEIMAGE

The **X11_MOVEIMAGE** function move image buffer data into a local buffer in real mode (i.e. below 1MB) space.

Entry	Description	Value
AH	Function number	X11_MOVEIMAGE
AL	Device driver/board number	First board in system 0, second 1, etc.
BL	Frame number	
BH:CX	Offset within image buffer	Offset in bytes.
DX	Number of bytes to move	1 to 65536 bytes can be moved. Set DX to zero for a 65536 byte move.
SI:DI	Pointer to destination buffer	Segment:offset pair pointer to real mode buffer.
<hr/>		
Exit	Description	Value
AX	Completion status	Zero if successful. Error code if failed.

4.21 X11_GETFRAMEADDR

The **X11_GETFRAMEADDR** function obtains the physical address of a particular frame.

Entry	Description	Value
AH	Function number	X11_GETFRAMEADDR
AL	Device driver/board number	First board in system 0, second 1, etc.
BX	Frame number	

Exit	Description	Value
AX	Completion status	Zero if successful. Error code if failed.
DX:CX	Frame physical address	Flat 32-bit address

Note: This function is used to access the frame buffer directly in protected mode. DPMI functions may be of assistance.

4.22 X11_WRITEPORT

The **X11_WRITEPORT** function writes to the 4-bit output port of the frame grabber.

<u>Entry</u>	<u>Description</u>	<u>Value</u>
AH	Function number	X11_WRITEPORT
AL	Device driver/board number	First board in system 0, second 1, etc.
BL	Data to write to the output port	Only lower 4 bits are significant
DL	Write mask	A value of 1 allows modification of a corresponding bit. Facilitates modifications of individual bit(s) without affecting the others.
<u>Exit</u>	<u>Description</u>	<u>Value</u>
AX	Completion status	Zero if successful. Error code if failed.

4.23 X11_READPORT

The **X11_READPORT** function reads the 4-bit input port of the frame grabber.

Entry	Description	Value
AH	Function number	X11_READPORT
AL	Device driver/board number	First board in system 0, second 1, etc.
Exit	Description	Value
AX	Completion status	Zero if successful. Error code if failed.
BL	Data read from the input port	Only lower 4 bits are significant

5.0 Display Functions

The display adapter must be VESA 2.0 compatible for any of the display functions to work. The degree of compatibility varies between display adapter manufactures and as a result the functions may not work reliably and/or selected modes may not function correctly.

In order to use these functions and have a properly formatted image on the display, the color selected in the **MODE** structure (see mode.color) must correspond to the video adapter's color format.

5.1 X11_DISPOPEN

The **X11_DISPOPEN** sets the display to a particular VESA display mode. This is the first video display function to be called for each frame grabber using the display functions.

<u>Entry</u>	<u>Description</u>	<u>Value</u>
AH	Function number	X11_DISPOPEN
AL	Device driver/board number	First board in system using display functions.
BX	Display mode	VESA mode number. See video adapter documentation for possible modes
<hr/>		
<u>Exit</u>	<u>Description</u>	<u>Value</u>
AX	Completion status	Zero if successful. Nonzero if failed.

5.2 X11_DISP

The **X11_DISP** function copies the contents of an image frame to the video adapter memory at specified screen coordinates and video buffer page.

<u>Entry</u>	<u>Description</u>	<u>Value</u>
AH	Function number	X11_DISP
AL	Device driver/board number	First board in system 0, second 1, etc.
BX	Frame number	
CX	x coordinate in pixels	
DX	y coordinate in pixels	
<u>Exit</u>	<u>Description</u>	<u>Value</u>
AX	Completion status	Zero if successful. Nonzero if failed.

Note: If the image is larger than or equal to the display resolution the display origin is set to 0,0 and putting values in CX,DX will have no effect.

Note: The user can expect higher performance if the image is larger than or equal to the display resolution. This is a result of being able to dump the image directly into the video buffer with little overhead.

Note: The video buffer paging can be used to write to the video buffer while displaying another area of the video buffer. The page granularity is 64KB.

5.3 X11_PUTPIXEL

The **X11_PUTPIXEL** function sets a specified pixel in the video buffer to a given color.

Entry	Description	Value
AH	Function number	X11_PUTPIXEL
AL	Device driver/board number	First board in system 0, second 1, etc.
CL	8bit color value (1 byte/pixel)	Depends on image <i>mode.color</i>
CX	16bit color value (2 bytes/pixel)	
DL:CX	24bit color value (3 bytes/pixel)	
DX:CX	32bit color value (4 bytes/pixel)	
BX	x coordinate in pixels	
DI	y coordinate in pixels	
Exit	Description	Value
AX	Completion status	Zero if successful. Nonzero if failed.

Note: The video buffer paging can be used to write to the video buffer while displaying another area of the video buffer. The page granularity is 64KB.

5.4 X11_GETTPIXEL

The **X11_GETTPIXEL** function gets a specified pixel in the video buffer and returns its color.

Entry	Description	Value
AH	Function number	X11_GETTPIXEL
AL	Device driver/board number	First board in system 0, second 1, etc.
BX	x coordinate in pixels	
DI	y coordinate in pixels	
Exit	Description	Value
AX	Completion status	Zero if successful. Nonzero if failed.
CL	8bit color value (1 byte/pixel)	Depends on image <i>mode.color</i>
CX	16bit color value (2 bytes/pixel)	
DL:CX	24bit color value (3 bytes/pixel)	
DX:CX	32bit color value (4 bytes/pixel)	

5.5 X11_DISPCLOSE

The **X11_DISPCLOSE** closes the display system and restore the previous display mode.

Entry	Description	Value
AH	Function number	X11_DISPCLOSE
AL	Device driver/board number	First board in system using display functions.

Exit	Description	Value
AX	Completion status	Zero if successful. Nonzero if failed.

Note: This function is called only once per application regardless of the number of frame grabbers installed and must be called with the same frame grabber number as **X11_DISPOPEN** was called.