ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Find the *RadiSys EXM-14* at our website: *Click HERE*

EXM-14
Live Motion Video EXM
Hardware Reference

RadiSys® Corporation

15025 S.W. Koll Parkway

Beaverton OR 97006

(503) 646-1800

FAX: (503) 646-1850

07-0094-02 August 1994

EPConnect is a trademark and EPC and RadiSys are registered trademarks of RadiSys Corporation. OS/2, IBM, and PC/AT are trademarks of International Business Machines Corporation. Microsoft and MS-DOS are registered trademarks of Microsoft Corporation.

November 1991

Copyright © 1991, 1994 by RadiSys Corporation

All rights reserved.

Hardware Warranty

RadiSys Corporation ("RadiSys") warrants the EPC system and component modules to the original purchaser for two years from the product's shipping date. If any RadiSys product fails to operate in compliance with its specification during this period, RadiSys will, at its option, repair or replace the product at no charge. The customer is, however, responsible for shipping the product; RadiSys assumes no responsibility for the product until it is received. This warranty does not cover repair of products that have been damaged by abuse, accident, disaster, misuse, or incorrect installation.

RadiSys' limited warranty covers products only as delivered. User modification, such as the addition of memory arrays or other devices, may void the warranty, and if the product is damaged during installation of the modifications, this warranty does not cover repair or replacement.

This warranty in no way warrants suitability of the product for any specific application.

IN NO EVENT WILL RADISYS BE LIABLE FOR ANY DAMAGES, INCLUDING LOST PROFITS, LOST SAVINGS, OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PRODUCT EVEN IF RADISYS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY PARTY OTHER THAN THE PURCHASER.

THE ABOVE WARRANTY IS IN LIEU OF ANY AND ALL OTHER WARRANTIES, EXPRESSED OR IMPLIED OR STATUTORY, INCLUDING THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR USE, TITLE AND NONINFRINGEMENT. Repair or replacement as provided above shall be the Purchaser's sole and exclusive remedy and RadiSys' exclusive liability for any breach of warranty.

NOTES

Table of Contents

1. Product Description....1

Specifications....2

2. Installation....3

Insertion in an EXM Carrier....3
Configuration Using BIOS Setup Screen....4
Enabling the EXM module....4
Connecting the Video Input Source....6
Connecting a Monitor....7

3. Connectors....9

Connectors....9

4. Programming Interface....13

Configuration Registers....13
I2C access....14
Analog Input Control....14
Composite Decoder Parameters....15
DCSC Registers (SAA7192)....16

5. Theory of Operation....17

Functional Overview....17
Hardware Overview....20
Video Input....21
Frame Buffer....22
Video Output....22
Coordinate Systems....23
Three Coordinate Spaces: VI, FB, and S....24
Map Frame Buffer to Screen....24
Map Video Input to Frame Buffer....25
Examples: Exact fit, Pan, Scale....26
Configuring and Using the PC Video Functions....27

EXM-14 Configuration 27
GUI Examples 28
PCVTUNER 28

6. PC Video Functions 31

PCV_Display 31
PCV_FillRect 32
PCV_Freeze 33
PCV_GetAcqWinMap 33
PCV_GetColorKey 34
PCV_GetComposite 35
PCV_GetContrast 36
PCV_GetRGB 36
PCV_GetScrWinMap 37
PCV_GetState 37
PCV_ReadRect 38
PCV_SetAcqWinMap 39
PCV_SetColorKey 41
PCV_SetComposite 42
PCV_SetCompositeLevel 43
PCV_SetContrast 43
PCV_SetRGB 43
PCV_SetScrWinMap 44
PCV_SetState 45
PCV_WriteRect 46
Supported Environments 47
Windows 47
OS/2 49

7. Support and Service 51

In North America 51
Technical Support 51
Bulletin Board 51
Repair Services 52
Warranty Repairs 52
Non-Warranty Services 52
Arranging Service 53
Other Countries 54

Appendix A: Video Standards.....A-1

Appendix B: Glossary B-1

Index I-1

Figures

Figure 1. EXM Setup Screen 4
Figure 2. EXM Slot Numbering..... 5
Figure 3. Hardware Overview..... 20
Figure 4. Mapping From Frame Buffer to Screen Coordinates 25
Figure 5. Mapping From Video Input Coordinates to Frame Buffer 25

Tables

Table 1. EXM-14 Environmental and Electrical Specifications 2
Table 2. OB1 Configuration Bit Definition 6
Table 3. DB-15 Connector Pin-out 9
Table 4. Auxiliary Connector Pin-out..... 10
Table 5. Frame Buffer Space 24

1. Product Description

This manual contains the information you need to install and use the EXM-14 Live Motion Video controller and its accompanying software diskette.

The EXM-14 is a video acquisition board that allows live video from one of several types of sources to be displayed and manipulated on a VGA graphics screen. It also allows images to be frozen, stored, and recalled.

The EXM-14 uses the Chips and Technologies 82C9001 PC-video chip to provide control of the video data. The data is stored in 15-bit RGB format which provides 32,768 colors to be displayed.

The EXM-14 hardware contains an EXM-bus interface, the C&T chip, a 1MB frame buffer, video output and multiplexing, A-D conversion, and a digital composite decoder circuit.

The EXM-14 is used in concert with an EXM-13/EXM-13A VGA graphics controller. It connects to the EXM-13/EXM-13A through the EXM-13/EXM-13A feature connector.

The EXM-14 cannot be used with products that do not have two EXM slots such as the EPC-6 or with products that have integrated VGA controllers such as the EPC-2 and EPC-3.

The software supplied with the EXM-14 provides device drivers and an application programming interface library to use the EXM-14 with either Microsoft Windows or with OS/2 Presentation Manager. An application is provided that displays full-motion video in a window. Sample code is provided to show how to use the application programming interface to the EXM-14 in a Windows or Presentation Manager application.

RadiSys ships a cable with the EXM-14 that connects the EXM-14 to the EXM-13A.

1 1

Specifications

The following table defines the power and environmental specifications of the EXM-14.

Characteristic		Value
Temperature	operating	0 to 60°C ambient
	storage	-40 to 85°C
Humidity	operating	5 - 90% noncondensing
	storage	8 - 95% noncondensing
Altitude	operating	10,000 ft (3000 m)
	storage	50,000 ft (15,000 m)
Vibration	operating	0.015 inch (0.38 mm) P-P displacement with 2.5 g peak (max) acceleration over 5-2000 Hz
	storage	0.030 inch (0.76 mm) P-P displacement with 5.0 g peak (max) acceleration over 5-2000 Hz
Shock	operating	30 g, 11 ms duration, half-sine shock pulse
	storage	50 g, 11 ms duration, half-sine shock pulse
Power	maximum	16 W
	typical	9 W
Current	maximum	+5V @ 2.3A +12V @ 260mA
	typical	+5V @ 1.2A +12V @ 200mA
Weight (EXM-14)		5.2 oz (150 g)
Weight (cable)		4.2 oz (120 g)

Table 1. EXM-14 Environmental and Electrical Specifications.

2. Installation

WARNING!

DO NOT PLUG IN ANY CABLE OR CONNECTOR INTO THE FRONT PANEL CONNECTORS WHILE THE EXM-14 IS POWERED UP. THIS PRODUCT IS NOT DESIGNED FOR "HOT INSERTION" WHILE THE SYSTEM IS POWERED UP, AND IS NOT DESIGNED TO WITHSTAND POTENTIAL DAMAGE THAT COULD ARISE FROM FLUCTUATIONS IN POWER. NEVER PLUG IN A SERIAL OR PARALLEL DEVICE, KEYBOARD, TRANSCEIVER, OR OTHER COMPONENT WHILE THE SYSTEM IS ON.

Before installing your EXM-14, you should unpack and inspect it for shipping damage.

- **DO NOT REMOVE THE MODULE FROM ITS ANTI-STATIC BAG UNLESS YOU ARE IN A STATIC-FREE ENVIRONMENT. THE EXM-14, LIKE MOST OTHER ELECTRONIC DEVICES, IS SUSCEPTIBLE TO ESD DAMAGE. ESD DAMAGE IS NOT ALWAYS IMMEDIATELY OBVIOUS, IN THAT IT CAN CAUSE A PARTIAL BREAKDOWN IN SEMICONDUCTOR DEVICES THAT MIGHT NOT IMMEDIATELY RESULT IN A FAILURE.**

- **ENSURE THAT THE INSTALLATION PROCESS AS DESCRIBED HEREIN IS ALSO PERFORMED IN A STATIC-FREE ENVIRONMENT.**

Insertion in an EXM Carrier

Insertion of the EXM-14 into an EXM carrier is straightforward. Remove any blank EXM panel from the carrier (by unscrewing the thumbscrews) and insert the EXM-14 into the card guides. Firmly press the EXM-14 front panel to ensure that it has mated with the rear connector, and secure it with the thumbscrews. DO NOT OVERTIGHTEN.

2 2

- **MAKE SURE THAT POWER TO YOUR SYSTEM IS OFF. THE EXM IS NOT DESIGNED TO BE INSERTED OR REMOVED FROM A LIVE SYSTEM.**

- **WHEN INSERTING THE EXM, AVOID TOUCHING THE CIRCUIT BOARD, AND MAKE SURE THE ENVIRONMENT IS STATIC-FREE.**

Configuration Using BIOS Setup Screen

The EXM configuration data in the EPC to which the EXM-14 is connected needs to be modified to recognize and enable the card and select from the available options. Invoke the BIOS setup function by pressing the CTRL-ALT-ESC keys simultaneously.

Enabling the EXM module

Once in the setup program, a menu displays specifying which function keys are available for further configuration. Press the F2 function key to invoke the EXM menu. The screen display resembles Figure 1 below.

```
          ID OB1 OB2
Slot 0    FF  00  00
     1    F5  05  00
     2    F1  01  00
     3    ED  01  00
     4    DE  01  00
     5    7D  05  91
```

Figure 1. EXM Setup Screen.

The setup screen displays the EXM configuration data (in hexadecimal) stored in nonvolatile memory which the EPC uses at power-up to recognize and configure each installed EXM. The displayed data shows SLOT, ID, OB1 and OB2 for each installed EXM.

All slots **not** occupied by an EXM module should show an ID of FF and OB1/OB2 of 00 00 indicating that no EXM is present. A typical value for the EXM-14 is shown in bold letters in slot 2.

Slot, OB1 and OB2 are defined as follows:

SLOT	indicates the slot in which the EXM is installed. See the diagram below to determine which EXM slot the EXM-14 occupies. Note that dashed lines indicate EXM slots that may not be available on all systems.

EXM Slots

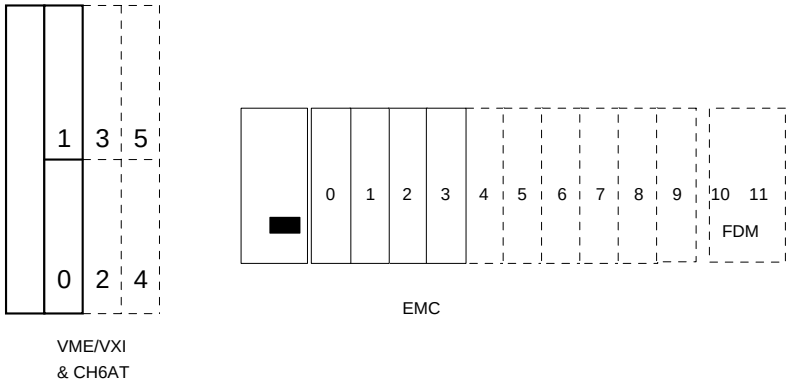


Figure 2. EXM Slot Numbering.

ID	is a hard-wired ID value. Each EXM has a unique ID value.
OB1/OB2	are two bytes of option information.

To add or change an EXM configuration, use the cursor keys (arrows) to move between the fields on the screen. Move the cursor to the appropriate slot entry and type in the correct value.

The **ID** for the EXM-14 is **F1**h.

2 2

OB1 and OB2 are two bytes of option information. **OB1** is a hexadecimal value derived by combining the following:

Reserved (bits 7-3)	I/O Base (bits 2-1)	Card Enable (bit 0)
00000	00 - AD6h (2D6h) 01 - AE6h (2E6h) 10 - BA6h (3A6h) 11 - BE6h (3E6h)	0 - disabled 1 - enabled

Table 2. OB1 Configuration Bit Definition.

Bits 2 and 1 are used to set the I/O base address.

For instance, a typical OB1 value is **01**h (00000001), representing card enabled, and an I/O address of AD6h.

OB2 is not used and should be set to **00**.

The EXM-14 also occupies 1 MB of memory space above 1M, at a base address that is programmable through the I/O space. If using RadiSys software, this address is set to 16 Mbytes. Otherwise, refer to Chips & Technologies spec sheet for the 82C9001 chip.

Connecting the Video Input Source

The cable shipped with the EXM-14 provides the interconnections with an EXM-13/EXM-13A, EXM-14, and the external video source.

This cable has two high-density 36-pin connectors and 4 female BNC jacks.

The high-density connector which has the four coaxial cables terminated in BNC jacks is plugged into the EXM-14. The other high-density connector is plugged into the EXM-13/EXM-13A. The connectors are keyed and cannot be inserted backwards. If the connectors are plugged into the wrong EXM's (i.e., the EXM-14 end into the EXM-13/EXM-13A) no damage will be done to the EXM's.

The four BNC jacks are labeled one of two ways. They may have text labels, which read RED, GRN/CHR, BLUE, and COMP/SYNC. Alternately, they may have red, green, and blue heatshrink tubing applied to the cables just above the connectors.

For composite video applications, the video is applied to the COMP/SYNC connector and the others are unused. For super VHS, the luma channel is applied to the COMP/SYNC connector, and the chroma is connected to GRN/CHR. For RGB video with separate sync, the red, green, blue, and sync signals are connected to the RED, GRN/CHR, BLUE, and COMP/SYNC connectors respectively.

Connecting a Monitor

A monitor is connected via the 15-pin D connector on the EXM-14 front panel. This connector provides the output circuit to the VGA display that would normally be provided by the 15-pin D connector on the EXM-13/EXM-13A when using the EXM-13/EXM-13A without the EXM-14. The 15-pin D connector on the EXM-13/EXM-13A should be left unconnected. The EXM-14 produces analog color signals, meaning that old-style EGA, CGA, and monochrome monitors cannot be used. VGA monitors may be used. If using a multiscan monitor, make sure to set the monitor's switch to analog (not TTL).

Monitors that can be used are VGA-compatible monitors (i.e., those compatible with the IBM PS/2 and with PC VGA add-in cards) and multiscan (multifrequency or "multisync") monitors. If you cannot mate your monitor to the EXM-14 because you have a cable with a 9-pin connector, either (1) you have a TTL monitor that is not compatible with VGA or (2) you have a multisync monitor (which are usually shipped with 9- and 15-pin cables or adapters) and are using the wrong cable.

To ensure a reliable connection, the monitor's cable should be screwed into the EXM-14's connector.

The EXM-14 produces good video results when the EXM-13/EXM-13A VGA controller is operated in either 640 x 480 or 800 by 600 resolution modes. When operated in 800 by 600 mode, the live video portion of the output does not fill the entire screen. Thus, in a Windows environment, one can have a full live motion display in a portion of the screen while other Windows are still visible. In 640 by 480 mode, the live motion video window can fill the entire screen. Refer to the EXM-13/EXM-13A reference to determine how to change VGA video modes.

NOTES

3. Connectors

Connectors

The DB-15 monitor connector is defined as the following:

Pin	Signal	Pin	Signal
1	Red	9	(key)
2	Green	10	Digital Ground
3	Blue	11	(not used)
4	(not used)	12	(not used)
5	Digital Ground	13	Horizontal sync
6	Analog Ground	14	Vertical sync
7	Analog Ground	15	(not used)
8	Analog Ground		

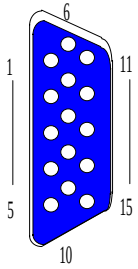
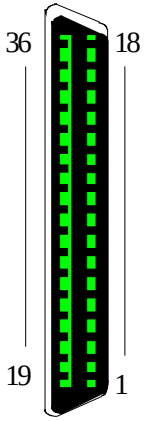


Table 3. DB-15 Connector Pin-out.

The EXM-14 contains a 36-pin auxiliary connector. This connector is a shielded female AMP 174726-3. It mates with an AMP 174340-3 male connector with 174747 shield case kit, or equivalent.

The pins on the auxiliary video connector on the EXM-14 are defined below.

Note that the EXM-13 and EXM-13A have identical pin-outs. The EXM-13A uses a blade connector, while the EXM-13 uses pins.



3 3

Pin	Signal	Pin	Signal
1	(not used)	19	Ground
2	P7	20	P0
3	Ground	21	P1
4	DCLK	22	P2
5	(not used)	23	Ground
6	P6	24	P3
7	Ground	25	P4
8	-BLANK	26	P5
9	HSYNC	27	Ground
10	VSYNC	28	(not used)
11	Ground	29	(not used)
12	Red (live video)	30	Analog Ground (for analog video in)
13	Green (live video)	31	Analog Ground (for analog video in)
14	Blue (live video)	32	Analog Ground (for analog video in)
15	COMP/SYNC	33	Analog Ground (for analog video in)
16	RED (VGA)	34	Analog Ground (for analog in from VGA)
17	GREEN (VGA)	35	Analog Ground (for analog in from VGA)
18	BLUE (VGA)	36	Analog Ground (for analog in from VGA)

Table 4. Auxiliary Connector Pinout.

The signals on the auxiliary connectors of the EXM-14 are described below.

-BLANK — This input signal is an alternate blanking signal from the EXM-13/EXM-13A.

BLUE (VGA) — This analog input is the blue color from the EXM-13/EXM-13A.

BLUE (live video) — This analog input is the blue color from the live video RGB source.

DCLK	This input signal is the dot clock from the EXM-13/EXM-13A.
GREEN (VGA)	This analog input is the green color from the EXM-13/EXM-13A.
GREEN (live video)	This analog input is the green color from the live video RGB source.
HSYNC	This input signal is the horizontal sync signal from the EXM-13/EXM-13A.
P0-P7	These input signals are the digital video inputs from the EXM-13/EXM-13A
RED (VGA)	This analog input is the red color from the EXM-13/EXM-13A.
RED (live video)	This analog input is the RED color from the live video RGB source.
COMP/SYNC	This analog input contains the composite video input of the live video source or it contains the sync information for RGB input.
VSYNC	This input signal is the vertical sync signal from the EXM-13/EXM-13A.

NOTES

3 3

4. Programming Interface

Configuration Registers

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	I/O port
Device ID Reg	1	1	1	1	0	0	0	1	100
Config Option Byte 1 Reg						I/O Base		Cden	102

These are standard EXM registers for device identification and configuration. The EXM-14 responds to accesses to these ports only if EXMbus line -EXMID is asserted. An 8-bit read from I/O address 100h returns the value F1, the device ID of the EXM-14. A read/write configuration register appears at I/O address 102h.

CDEN specifies whether the EXM is disabled or enabled. If disabled, the EXM will not respond to any I/O or memory addresses; it will only respond to reads from I/O port 100h and reads and writes from I/O port 102h, and then only if EXMbus line -EXMID is asserted.

I/O Base specifies the base address of the I/O address space to which the EXM-14 will respond according to the following table:

00	AD6h	(2D6h)
01	AE6h	(2E6h)
10	BA6h	(3A6h)
11	BE6h	(3E6h)

Addresses in parentheses are the addresses that cards that decode only A0-A9 may conflict with. Most EXMs fall into this category.

Note that the PCVideo chip requires that the lower 8 I/O address bits be programmed to the appropriate value. This may be done by blindly writing the lower 8 bits of the selected address to the address range defined by the I/O Base field in the configuration register whenever the software is invoked, before any other register access is attempted.

To assert the EXMID for a particular slot, one needs to write the slot number to the slot register (I/O register 96h). For instance, if the EXM-14 resides in slot 2 of the EXMbus subplane, then before accessing register 100h or register 102h, one should first write the value 2 to register 96h. When finished, it is good programming practice to restore the slot register with the value FFh so that other SW which manipulates these registers will not cause unpredictable behavior.

4 4

I^2C access

Three I/O bits are used to access the I^2C bus, which is used to communicate with some of the peripheral IC's. These bits are manipulated under software control. For information, refer to the Signetics I^2C specification. The I^2C bus is controlled by the General Purpose I/O Control register in the PC Video Chip (R18). Addresses for devices on the I^2C bus are as follows:

SAA7191 DMSD	8Ah
SAA7192 DCSC	E0h
TDA8444 DAC	48h

Analog Input Control

Image digitization is done in Signetics TDA8709 IC's (for RGB mode, and chroma in SVHS mode) or a TDA8708 IC (for composite signals). The gain of the TDA8709's can be adjusted for RGB mode by way of a DAC. Composite conversion is accomplished without the need for gain adjustment by an AGC circuit in the TDA8708.

RGB Video

For RGB data, the blanking level (0 IRE) is defined in EXM-14 to be a value of 16, and peak white (100 IRE) is defined to be a value of 235.

The blanking level will be set automatically by the TDA8709 IC's. The peak white level is set by programming the gain of the TDA8709's.

The TDA8444 is accessed via the I^2C bus. DAC0 corresponds to the red channel, DAC1 for green (which is also used for the chroma input for SVHS formats), and DAC2 for blue. DAC3-7 are unused.

The minimum input voltage to achieve a full-scale video signal is 0.40V P-P, which is achieved with the gain DAC set to maximum (63 decimal). The maximum voltage accepted without saturation is 1.77V P-P, with the gain DAC set to 0.

For RS-170 video inputs (1.0V P-P), the gain DAC should be set to midpoint (31 decimal). For RS-343 (0.707V P-P), a value of 43 decimal should be used.

For standard video formats (i.e., RS-170 or RS-343), the value that is written to all three DAC's will be equal, and constant. For non-standard levels or for "image enhancement" these values may be adjusted.

For SVHS Chroma input, the green/chroma channel (DAC1) should be set to 29 decimal.

To input RGB video, bit GPS1 of the SAA7191 must be set to 1.

Composite video

For composite video, input levels are set by an AGC (automatic gain control) circuit within the TDA8708 IC. Once the timing setup is correct no adjustments are needed.

To input composite video, bit GPS1 of the SAA7191 must be set to 0.

Composite Decoder Parameters

The composite decoder chips are accessed via the I^2C bus. One of the chips provides the decoding functions, and the second converts the luminance-chrominance output of the first to RGB.

DMSD (SAA7191)

This IC (Digital Multi-Standard Decoder) takes digitized composite video from the ADC's and decodes it (from whatever format, PAL, NTSC, SVHS, or SECAM) to a luminance/chrominance (Y/C) format. The DMSD also provides sync signals and clocks for the RGB mode of operation.

The DMSD must be programmed with values indicating the video standard being decoded, as well as parameters such as color saturation and hue. The DMSD can also be programmed to automatically decide what video standard is being decoded.

4 4

DCSC Registers (SAA7192)

The DCSC (Digital Color Space Converter) converts the Y/C data from the DMSD to digital RGB. It also may be used to do color transformations on RGB data.

DCSC contains three 256x8 RAM's that can be used to provide color correction. These are lookup tables, one location for each possible entry. For general use (normal mode), they may be bypassed by setting the GPSW2 bit of the DMSD to 1. Enabling the lookup tables (VLUT's) allows colors to be re-mapped; (i.e., green would appear red, negative image, etc.)

The DCSC can operate in a number of modes. In the EXM-14 application it is set to Format 6 (normal) or 8 (enhanced) for composite input operation, and is set to mode 9 (normal) or mode 11 (enhanced) for RGB operation. Note that normal mode is differentiated from enhanced mode by the state of the VLUTBYPASS pin, which is controlled by the DMSD output GPSW2.

5. Theory of Operation

The EXM-14 is a video acquisition board that allows full-motion video from one of several types of sources to be displayed and manipulated on a VGA screen. The EXM-14 hardware converts the full-motion video image into a frame buffer. The frame buffer can be programmed to capture full-size video images, cropped images, and scaled images. The frame buffer image is then merged with the standard VGA output to create a mixed video image to display on a VGA monitor. The frame buffer can be merged with the VGA controller's output at programmable X-Y coordinates or by color keying to a specified VGA color (or both).

This chapter describes the software that is supplied with the EXM-14. Each function in the application programming interface is individually listed and explained. The environmental requirements for developing and running EXM-14 applications are listed.



Functional Overview

The video input circuit accepts the video in either RGB or composite form and digitizes it.

RS-170/RS-343 RGB video input formats provide separate signals for Red, Green, and Blue. Sync signals are provided as a separate composite sync signal.

The format of these signals is well defined by the standards RS-170 and RS-343. (The difference between these standards is a small level difference, plus higher resolution in RS-343). The RGB format is preferred over composite video signals because of its higher image quality and bandwidth, and is used by most industrial imaging systems.

The RGB video input circuit uses 3 flash A-D converters to digitize the RGB analog signals. The gain of these signals is set by a Signetics TDA8444 A-D converter.

The sync signal is processed by the Signetics composite decoder IC's. These IC's generate the clocks for the rest of the input circuitry. The input video timing must correspond to one of the broadcast video timing systems that the decoder supports. Normally for RGB video, this will be RS-170.

Additionally, an LM1881 sync separator IC is provided to generate a FIELD signal for the PC Video chip in RGB mode.

The Composite video input circuit on the EXM-14 is designed to accommodate all of the composite video standards, including SVHS (Super VHS, a separate chroma/luma system). NTSC, PAL, and SECAM are broadcast video standards used in North America (and Japan), greater Europe, and France, respectively. There are minor variations of each in individual countries. RS-170A is a closed-circuit standard comparable to the NTSC broadcast standard.

All composite video formats are supported. However, color information is decoded only for NTSC/M, PAL/B, PAL/G, PAL/H, and SECAM formats. Other formats are supported in monochrome only.

In a composite video signal, both luminance (intensity), chrominance (color), and sync are encoded into one analog signal. (This is done to allow broadcasting over a single carrier, such as a TV transmitter). In order to put all this information into the limited bandwidth of a TV channel, the information is limited in its bandwidth. For the NTSC standard, luminance data is limited to less than 3.5 MHz, and chrominance information is carried by the phase of a 3.57 MHz subcarrier. Thus, the image quality of composite video suffers when compared to RGB.

SVHS video is similar to other composite video standards, except that the chroma information is transmitted in a separate cable from the luminance and sync information. This allows much higher bandwidth.

Common sources of composite video include TV cameras, camcorders, and VCR's.

The composite signal is digitized by a separate flash A-D converter, which has Automatic Gain Control to accommodate variations in input levels. The digital composite video is then processed by Philips SAA7191 multistandard decoder (DMSD) and SAA7192 color space converter IC's into digital RGB.

In SVHS video, the chroma channel is digitized by a separate flash converter and is then fed to the DMSD.

The sync signal is processed thru the SAA7191 DMSD IC.

The frame buffer consists of 8 256Kb x 4 100nS Video Rams.

Data is written to the memories synchronous with the incoming video pixel clock (generated by the DMSD), under control of the PC Video chip.

Data is read from the memories through the VRAM's shift-register outputs, alternating the high-byte and low-byte data. Read data is synchronized with the Horizontal Sync, Vertical Sync, and Dot Clock signals from the EXM-13/EXM-13A VGA feature connector.

The control circuit is implemented almost entirely in the Chips & Technologies PC-video chip, the 82C9001A. This IC provides the registers and interface to the EXM-bus, the logic required to do scaling and windowing (scaling is done by dropping pixels) and the keying of the incoming video over the VGA video.

Keying of the live video over the VGA video can be accomplished in two ways by the PC Video chip. In the first, a window is created by timing; i.e., a certain number of lines are counted from Vertical Sync, and a certain number of pixels from Horizontal Sync.

The second method is similar to color-keying done in broadcasting. The digital video data from the VGA controller is compared to a color key, and if the data matches the video source is switched from the VGA controller to the live video. This allows a video window of arbitrary shape and size.

The video output circuit takes the video from the frame buffer and converts it to analog RS-343 video by a triple video DAC.

These analog RGB signals are then multiplexed with the analog RGB signals from the external VGA controller. The control for this mux is provided by the logic in the PC Video chip.

The output level of the EXM-14 is adjustable from approximately 0.5V P-P to over 1V P-P. Adjustment is accomplished by using a channel of the TDA8444 DAC to adjust the full-scale reference of the DAC. This DAC in initialized to provide a level of approximately .707V, in accordance with the spec for RS-343 (without sync). This provides an adjustment of the picture contrast, and allows it to be matched to the VGA video if desired.

Hardware Overview

This section describes the EXM-14 hardware from the programmer's point of view. The video data acquisition and display process is described, showing how the hardware allows the application programmer to set up and then mostly ignore the EXM-14 board. The section discusses the video input and output parameters critical for acquiring and mixing video data. The frame buffer data format is also explained.

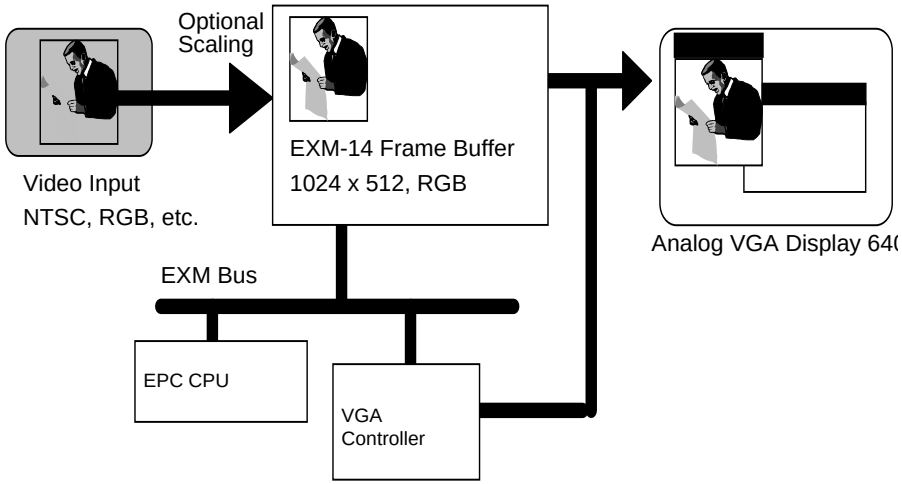


Figure 3. Hardware Overview.

Acquisition and Display Process

Video image acquisition and display are two independently programmable processes in the EXM-14. The acquisition of an image and the display of the image occur at asynchronous clock rates. The asynchronous processes are coordinated through dual-port memory accesses to the frame buffer. The acquisition and display processes are illustrated in Figure 3.

Full-motion video input comes into the EXM-14 through a connector on the EXM-14 front panel. You program the input process to specify the video type (e.g. NTSC, RGB, etc), the gain on the analog signals, and so on. This defines the video input image.

The video input image is converted to an RGB representation and stored into the frame buffer. You can specify the acquisition window coordinates to implement either cropping or panning of the video input image. You can also independently scale the horizontal and vertical dimensions of the video input image. Horizontal scaling is done by dropping pixels; vertical scaling is done by dropping scan lines. The cropped and scaled video input image is stored in the frame buffer.

The frame buffer RGB data is converted to analog format, multiplexed with the VGA controller's analog output, and then sent out through the VGA connector on the EXM-14 front panel. You program the output multiplexing process to specify the output window location, the color key to match, and what combination of these two conditions cause which video source to be output to the EXM-14 VGA connector. No scaling is done on the frame buffer data, so each pixel in the frame buffer image corresponds to a pixel in the VGA controller's image data.

Video Input

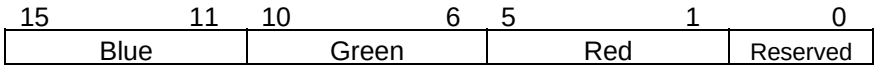
The EXM-14 accepts a broad range of video signal formats as input, converts those to a single RGB format, and stores that format in the frame buffer. The supported video input signals have three characteristics: the type, the timing information, and the level information. The programmer normally specifies only the type and level information before starting the video acquisition process. The EXM-14 automatically locks on to the signal.

The video input signal type is either composite or RGB. There are a variety of composite video standards. The EXM-14 supports the NTSC, PAL, and SECAM standards, and their standard modifications. Super VHS composite video is also supported. The standards for RGB video are not as tightly specified. The EXM-14 supports many non-standard specifications, as well as RS-170 and RS-343. RS-170 is NTSC timing with standard levels. RS-343 specifies levels but not the timing information.

The level information is specified differently for composite and RGB signals. The level information for RGB video signals specify the gain adjustments on the A/D converters located in front of the red, blue, and green analog input signals. The level information for composite video signals specifies the adjustment for the hue and saturation.

Frame Buffer

The frame buffer holds 512 scan lines consisting of 1024 pixels per scan line. Each pixel is stored as a 16-bit RGB value: 5 bits of red, 5 bits of green, 5 bits of blue, and 1 reserved bit that must be 0.



The frame buffer is dual ported. One port is dedicated to the output process, and is used when mixing the frame buffer image with the VGA controller image. The other port to the frame buffer is shared by the video input image data and the EPC processor. The video input must be frozen for the EPC processor to read or write the frame buffer.

The frame buffer is addressed as 1 MByte of memory, located on a (programmable) megabyte boundary in the EPC's extended memory space. In most cases, the 24-bit addressing limitation of the EPC running in 286 protected mode requires that the EPC not have more than 8 MBytes of system memory when using an EXM-14. This limitation can be worked around in systems that run 386 protected mode software (by addressing the memory at 16MB plus the frame buffer address). Also, this limitation can be ignored in systems where EPC access to the frame buffer is not required.

Some EPC processors cache memory accesses. On those processors, software is required to invalidate or flush any cached frame buffer data after suspending the data acquisition and before reading the frame buffer. This is done for the programmer by the PC Video software. If not using RadiSys software, refer to the C&T spec sheet for the 82C9001 chip.

Video Output

The EXM-14 hardware selects, on a pixel by pixel basis, whether frame buffer data should replace VGA supplied data for the video image that is sent out through the VGA connector on the EXM-14 front panel. For each pixel, the selection of which data to display depends on the X-Y location, the VGA pixel's color, and whether the frame buffer is enabled for display. The hardware is programmed to use combinations of these conditions to select the video output source. Generally, when the frame buffer is enabled for display, and the X-Y location is inside a programmed rectangular region, and the VGA pixel's color matches a programmed color, then the pixel from the frame buffer is displayed; otherwise, the VGA pixel is displayed.

The merging of the frame buffer data and the VGA screen data is programmed through registers in the EPC I/O space. These registers specify:

a) the upper left corner, width and height of a rectangular region of the VGA screen,
b) the upper left corner of the frame buffer that corresponds to the upper left corner of the rectangular window on the VGA screen,
c) a color mask and color match value to compare VGA pixel colors,
d) a flag to enable or disable the frame buffer as a video output source, and
e) miscellaneous video output control functions.

Coordinate Systems

This section describes three different video coordinate systems that an EXM-14 programmer will work with. These coordinate systems are the video input coordinate space, the frame buffer coordinate space, and the screen coordinate space. The first two coordinate spaces are private to the EXM-14, but the screen coordinate space is shared between the EXM-14 and the VGA controller.



In most cases, the EXM-14 programmer will be working within the context of other graphical coordinate spaces as well as those coordinate spaces recognized by the EXM-14. For example, in Microsoft Windows the GDI layer maps the logical window coordinate space into the device dependent coordinate space. To use the EXM-14 in this context, the programmer will need to obtain from the GDI routines the device dependent screen coordinates to set up the EXM-14 video window (and track changes to those screen coordinates as well).

This section emphasizes the relationships the programmer creates between the three coordinate systems by discussing the transformations between the coordinate spaces. The EXM-14 uses two independent coordinate space transformations: video input space to frame buffer space during the acquisition process, and frame buffer space to screen space during the video display process. Combined with direct CPU access to the frame buffer, these independent coordinate transformations allow the programmer to create some very sophisticated visual effects. This section and the PC Video software in general, however, concentrates primarily on the functional subset that maps complete video input frames into windows on the display screen.

Three Coordinate Spaces: VI, FB, and S

The EXM-14 works in three video coordinate spaces: the video input space *VI*, the frame buffer space *FB*, and the screen space *S*. Each of these coordinate spaces are rectangular. Each have their origin (0, 0) in the upper left corner, and extend to the point (H, V) as the lower right boundary outside the space. Although the values of H and V differ for each coordinate space, the origin and orientation of each coordinate system remains the same.

The first pixel clock after VSYNC is point (0, 0). The next pixel clock is point (1, 0). The rightmost valid point in VI space is (HsyncStart-1, 0) at pixel clock number HsyncStart-1. The next valid point in VI space is (0, 1), which occurs at pixel clock Hcount. The last valid point in VI space is (HsyncStart-1, NumScanLines-1).

The FB space is fixed by the EXM-14 hardware. Each pixel is a 16-bit word. There are 1024 pixels in the horizontal direction, and 512 pixels in the vertical direction.

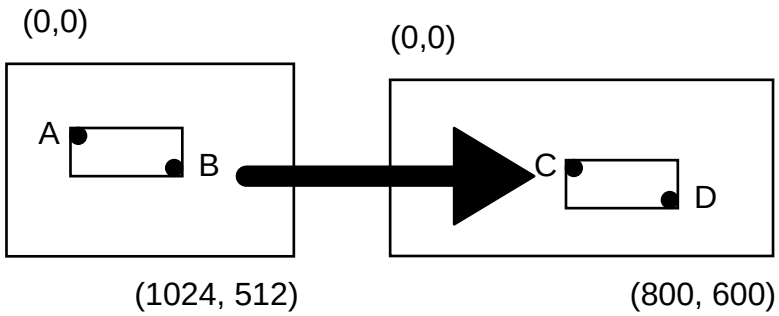
Frame Buffer Space		
Corner	**Coordinate**	**Byte offset**
Upper left	(0, 0)	0x00000
Upper right	(1023, 0)	0x007FE
Lower left	(0,511)	0xFF800
Lower right	(1023,511)	0xFFFFE

Table 5. Frame Buffer Space.

The S space is defined by the VGA controller. Depending on which text mode or graphics mode the VGA controller is currently programmed to, the S space can take on any number of H and V extents. The common graphics resolutions are 640 by 480, 800 by 600, and 1024 by 768.

Map Frame Buffer to Screen

The mapping between the frame buffer and the screen by the EXM-14 is completely determined by two points and the window size: the upper left corner of the frame buffer window, the upper left corner of the screen where the upper left corner of the frame buffer window maps to, and the size of the screen window. Figure 3 illustrates the mapping. Because the EXM-14 maps the frame buffer pixels directly to screen pixels, the lower right corner of the frame buffer window B is the point A+(D-C).



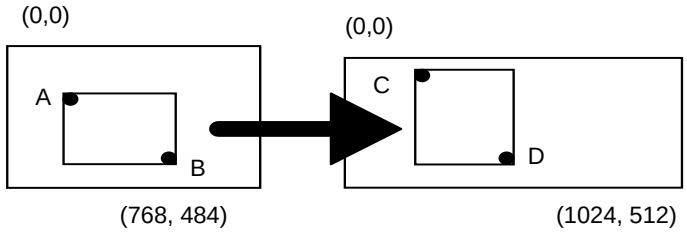
A: Upper left corner of FB window
B: Lower right corner of FB window
C: Upper left corner of S window
D: Lower right corner of S window

Figure 4. Mapping from Frame Buffer to Screen Coordinates.

The **PCV_SetScrWinMap** function sets up the arbitrary mapping between the frame buffer and the screen. The EXM-14 hardware requires points A and B to be within the FB space, and points C and D to be within the S space.

Map Video Input to Frame Buffer

The mapping between the video input and the frame buffer by the EXM-14 is determined by two points and two window sizes. The upper left corner of the video input window maps to the upper left corner of the frame buffer window. The lower right corner of the video input window maps implicitly to a point in the frame buffer based on horizontal and vertical scale factors. Figure 5 illustrates the mapping.



A: Upper left corner of VI window
B: Lower right corner of VI window
C: Upper left corner of FB region
D: Lower right corner of FB region

Figure 5. Mapping from Video Input coordinates to Frame Buffer

The **PCV_SetAcqWinMap** function sets up the arbitrary mapping between the video input and the frame buffer. The EXM-14 hardware requires points A and B to be within the VI space, and points C and D to be within the FB space.

Examples: Exact fit, Pan, Scale

By selecting the two different window mappings appropriately, the video input can be acquired and displayed in many different ways. Of these many ways, three particular mappings of the video input to the display screen are the most common:

1) an exact 1-to-1 display of the video input on the screen,
2) an exact 1-to-1 display of the video input in a smaller screen window that can be scrolled across the video input, and
3) a scaled display of the video input that fits the video input into a smaller screen window.



In the exact fit case, you start with the size of the video input frame. Assume that the frame has H horizontal pixels and V vertical pixels. Using **PCV_SetAcqWinMap**, set the video input window to (0, 0) through (H-1, V-1), and the frame buffer window to (0, 0) through (H-1, V-1). Next, locate a screen rectangle with H horizontal pixels and V vertical pixels. Assume the origin of this screen rectangle is (X, Y). Using **PCV_SetScrWinMap**, set the screen window to (X, Y) through (X+H-1, Y+V-1) and the frame buffer window to (0, 0) through (H-1, V-1). With this mapping, the entire video input frame will show up on the screen when video acquisition is enabled.

In the second case, where the screen window is smaller than the video frame, you map the video input window to the frame buffer in exactly the same way as the exact fit case. Now assume that the smaller screen window is M pixels wide and N pixels high, based at (X, Y). Then using **PCV_SetScrWinMap**, set the screen window to (X, Y) through (X+M-1, Y+N-1) and the frame buffer window to (0, 0) through (M-1, N-1). To pan across the video input frame by I pixels horizontally and J pixels vertically, again use **PCV_SetScrWinMap**, keeping the screen window constant and setting the frame buffer window to (I, J) through (I+M-1, J+N-1). You must take care to ensure that $0 <= I < H-M$ and $0 <= J < V-N$.

In the third case, where the video frame is scaled down to fit into the screen window, you start with the size of the screen window. Assume that the screen window is M pixels wide and N pixels high, based at (X, Y). Using **PCV_SetScrWinMap**, set the screen window to (X, Y) through (X+M-1, Y+N-1) and the frame buffer window to (0, 0) through (M-1, N-1). To fit the video frame to the screen window, set the last parameter of **PCV_SetScrWinMap** to a non-zero value.

Configuring and Using the PC Video Functions

This section describes how to configure and use the EXM-14 in a PC video application. Most of the programming complexity is in the initialization and configuration of the EXM-14 hardware for the particular video input and display format. After the input format has been selected and the coordinate space mappings established, the maintenance of the PC video data structures is minimal.

The software provided with the EXM-14 only works when the base address of the EXM-14 is set to AD6h.

EXM-14 Configuration

Before any programs will work with the EXM-14, the EXM configuration information for the board needs to be established. There are two parameters the user first sets: enable the board after power-on reset; and a non-conflicting I/O register base address (always AD6h unless loading a user-written driver not supplied by RadiSys). This information is stored in the non-volatile system configuration data.

Most of the initialization of the EXM-14 hardware is done by the PCVIDEO dynamic link library when it is loaded (and consequently requires no overt programming support). This initialization includes determining the current I/O register base, selecting a memory base for the frame buffer, and initializing the EXM-14 to a sane state. Among other things, this sane state includes freezing the frame buffer acquisition and disabling the display of frame buffer data on the VGA screen.

It is the programmer's responsibility to establish the basic video parameters for the application. For composite video, the following sequence of function calls is typical:

PCV_SetComposite(PCV_NTSC);
PCV_SetAcqWinMap(0, 0, 648, 484, 0, 0, 648, 484);
PCV_SetScrWinMap(0, 0, X, Y, M, N, TRUE);
PCV_SetColorKey(0x000F);
write screen rectangle (X, Y, M, N) to color 0x0F;
PCV_Freeze(FALSE);
PCV_Display(TRUE);

For RGB video input, replace the first function with a call to **PCV_SetRGB**.

GUI Examples

There are several common graphical user interface idioms that most EXM-14 applications will support. These idioms include moving the window to a different location on the screen, resizing the window to be smaller or larger, and minimizing the window to an icon. Each of these idioms are straightforward to implement with the PC video functions.

To move an existing window to a different location, you use **PCV_SetScrWinMap** and change the X and Y origin of the screen window. If the resulting window fits entirely on the screen, then you're done. However, if part of the window is moved off the screen, then you also need to adjust the M and N sizes of the window to reflect the true screen coordinate limits. When the window is moved back onto the screen, you will want to change M and N back again, so you'll need to keep track of or recompute the original M and N values.



To resize an existing window, you use **PCV_SetScrWinMap** and change just the M and N size parameters. You may want to keep track of what the frame buffer boundaries are, and not grow the window to be larger than those boundaries. If you're scaling the video input to fit it to the size of the window, then you'll also want to set the last parameter in the **PCV_SetScrWinMap** to a non-zero value.

There are two different approaches to take when minimizing and restoring an EXM-14 window. The simple approach is to use **PCV_Display** to disable the display, and then take the window away and display the icon. When the window is restored, you repaint the window with the color key, and then call **PCV_Display** to enable the display. An alternative approach is to hook into the icon display process, and display in the icon a miniature view of the EXM-14 video input. The miniature view is obtained by using the **PCV_SetAcqWinMap** function to reduce the input image to icon size, and then **PCV_SetScrWinMap** to relocate the miniature window over the icon (which is colored in the color key, of course). To restore the window to standard size, restore the original parameters to the **PCV_SetScrWinMap** and **PCV_SetAcqWinMap** functions after the window has been repainted with the color key.

PCVTUNER

The PC Video software is supplied with the GUI application PCVTUNER. This program allows the user to connect the video input source to the EXM-14 front panel and then just see what can be seen. The program has menu options to set up the video input, and to control the display of that input on the VGA screen.

Some EXM-14 users will not need to write any application code themselves because PCVTUNER will do everything they need with the EXM-14. However, even those EXM-14 users who need to write their own PC Video application will find PCVTUNER useful to determine video input parameters and to see how PC Video functions can be integrated into a GUI application.

One use of PCVTUNER is to determine the correct video input parameters. In some cases standard levels and timing parameters need to be adjusted based on peculiarities of the video input source. For non-standard video input sources, PCVTUNER can be used to experimentally determine what the best level and timing parameters are. The selected video input parameters are saved as part of the context of the PCVTUNER program, so that the next time the program is run, it starts out with those parameters. The video parameters are saved in a format that is easy to use if you will directly call **PCV_SetComposite** or **PCV_SetRGB** when writing your own application.

Installation procedures for DOS and OS/2 are provided below:

Windows Installation



[1] Create a new directory.

```
C>MKDIR \PCVIDEO
```

[2] Copy all the files under the A:\WINDOWS directory to the new directory.

```
C>XCOPY /S A:\WINDOWS \PCVIDEO
```

[3] In your **AUTOEXEC.BAT** set the lib path to point to this directory.

```
SET LIB=%LIB%;c:\pcvideo
```

OS/2 Installation

[1] Create two new directories.
C>MKDIR \PCVIDEO
C>MKDIR \PCVIDEO\SAMPLES

[2] Copy files under the **A:\OS2** directory to the new directory.
C>COPY A:\OS2 \PCVIDEO

[3] Copy files under the **A:\OS2\SAMPLES** directory to other directory.
C>COPY A:\OS2\SAMPLES \PCVIDEO\SAMPLES

[4] Copy files under the **A:\OS2\WLO\DLL** to the **C:\OS2\DLL** directory.
C>COPY A:\OS2\WLO\DLL C:\OS2\DLL

[5] In your **CONFIG.SYS** file:

a. Set IOPL to yes
IOPL=YES

b. Install the pcvideo.sys driver
DEVICE=C:\PCVIDEO\PCVIDEO.SYS

c. Set the lib path to point to \PCVIDEO directory.
SET LIB=%LIB%;c:\pcvideo

6. PC Video Functions

This chapter describes the functions that are available to you when programming the EXM-14. The functions are listed in alphabetical order. The chapter describes for each function the calling syntax, what the function does, what parameters are valid, and what values are returned.

Include the file PCVIDEO.H in your program to obtain access to all the function prototypes, defined constants, and defined data structures. Link your program with an IMPORTS PCVIDEO.DLL statement in your program's linker ".DEF" file to resolve the references to the EXM-14 functions.

An explanation of how to use the EXM-14 functions to build a full-motion video application that works within the context of Microsoft Windows or the OS/2 Presentation Manager is beyond the scope of this specification. However, the sample code supplied with the PC Video software will have working examples of how this is done.

PCV_Display

int PCV_Display(*fDisplay*)

This function enables and disables the display of the frame buffer on the VGA screen. The previous state of the *fDisplay* flag is returned.

Parameter	Type	Description
fDisplay	**int**	If *fDisplay* is zero, then the frame buffer is not displayed on the VGA screen. If *fDisplay* is nonzero, then the frame buffer is enabled for display on the VGA screen.

The frame buffer is displayed on the VGA screen according to the mapping set by the last call to **PCV_SetScrWinMap**. This function does not freeze or start the acquisition of video input into the frame buffer.

PCV_FillRect

int PCV_FillRect(*xFB*, *yFB*, *cx*, *cy*, *data*)

This function initializes a rectangular region of the frame buffer to a fixed value. Acquisition of the video input data is temporarily suspended while the function runs.

Parameter	Type	Description
xFB	**unsigned short**	X-position of the frame buffer rectangle's upper left corner in frame buffer coordinates.
yFB	**unsigned short**	Y-position of the frame buffer rectangle's upper left corner in frame buffer coordinates.
cx	**unsigned short**	Width of the frame buffer rectangle in frame buffer coordinates. The X-coordinate of the frame buffer rectangle's lower right corner is *xFB*+*cx*-1..
cy	**unsigned short**	Height of the frame buffer rectangle in frame buffer coordinates. The Y-coordinate of the frame buffer rectangle's lower right corner is *yFB*+*cy*-1.
data	**unsigned short**	The 16-bit frame buffer data.

This function works directly with the frame buffer. The function does not work with the video input's window into the frame buffer, nor with the VGA screen's window into the frame buffer.

The function verifies that the frame buffer rectangle coordinates lie within the physical frame buffer.

Return Value	Description
PCV_SUCCESS	The function completed without errors.
PCV_FBBOUNDS	Either (*xFB*, *yFB*) or (*xFB*+*cx*-1, *yFB*+*cy*-1) is outside the physical boundaries of the frame buffer.

Unlike the **PCV_ReadRect** and **PCV_WriteRect** functions, this function can initialize the entire frame buffer in one call.

PCV_Freeze

int PCV_Freeze(*fFreeze*)

This function enables and disables the acquisition of video input into the frame buffer. The previous state of the *fFreeze* flag is returned.

Parameter	Type	Description
fFreeze	**int**	If *fFreeze* is nonzero, then the acquisition of video input into the frame buffer is halted. If *fFreeze* is zero, then acquisition of video input into the frame buffer is enabled.

The acquired video input is stored in the frame buffer according to the mapping set by the last call to **PCV_SetAcqWinMap**. This function does not enable or disable the display of the frame buffer on the VGA screen.

PCV_GetAcqWinMap

void PCV_GetAcqWinMap(*fpxVI, fpyVI, fpcxVI, fpcyVI, fpxFB, fpyFB, fpcxFB, fpcyFB*)

This function returns the current video input region acquisition parameters. The function returns the position and size of the video input window, as well as the position and size of the acquired image in the frame buffer.

Parameter	Type	Description
fpxVI	**unsigned short far***	X-position of the video input window's upper left corner in video input coordinates.
fpyVI	**unsigned short far***	Y-position of the video input window's upper left corner in video input coordinates.
fpcxVI	**unsigned short far***	Width of the video input window in video input coordinates.
fpcyVI	**unsigned short far***	Height of the video input window in video input coordinates.

fpxFB	**unsigned short far***	X-position of the frame buffer window's upper left corner in frame buffer coordinates.
fpyFB	**unsigned short far***	Y-position of the frame buffer window's upper left corner in frame buffer coordinates.
fcxFB	**unsigned short far***	Width of the frame buffer window in frame buffer coordinates.
fpcyFB	**unsigned short far***	Height of the frame buffer window in frame buffer coordinates.

The values returned for the width and height of the frame buffer may not be the same values that were specified on the previous call to **PCV_SetAcqWinMap**. The values returned reflect the actual width and height being written into the frame buffer. A difference may arise because the EXM-14 implements horizontal scaling by dropping some number of video input horizontal pixels per 64 horizontal pixels, and implements vertical scaling by dropping some number of video input scan lines per 64 scan lines.

The function returns a value for every parameter that is not a null pointer. If the parameter is a null pointer, then that parameter is not returned.

PCV_GetColorKey

unsigned short PCV_GetColorKey( )

This function returns the current color key. The color key controls which pixels inside the VGA screen window display the frame buffer, and which pixels display VGA screen data.

Return Value	**Description**
0xFF00	A color key of 0xFF00 means that all pixels inside the VGA screen window will be replaced with the frame buffer data.
0xFFFF	A color key of 0xFFFF means that all pixels inside the VGA screen window will display the VGA screen data.

0x00*HH*	A color key in the range 0x0000 to 0x00FF means that those pixels inside the VGA screen window with a VGA data value equal to the low order eight bits *HH* will be replaced with the frame buffer data, and all other pixel values will display the VGA screen data.
0x*MMHH*	A color key with any but not all of the upper eight bits *MM* nonzero means that those pixels inside the VGA screen window with VGA data value equal to the low order eight bits *HH* when masked by *MM*. In this case, each upper byte bit acts as a mask, and suppresses the compare on those data bits for which a '1' is written to the corresponding lower byte bit position *HH*.

PCV_GetComposite

int PCV_GetComposite(*fpType, fpHue, fpSaturation*)

This function returns the type and level parameters for composite video input.

Parameter	Type	Description
fpType	**int far***	One of 3 types of composite video. Will be ORed with **PCV_SVHS** if composite video was initialized as **PCV_SVHS**.
fpHue	**unsigned char far***	Hue levels vary between 0 and 255.
fpSaturation	**unsigned char far***	Saturation levels vary between 0 and 255.

The function returns a value for every parameter that is not a null pointer. If the parameter is a null pointer, then that parameter is not returned.

Return Value	Description
PCV_SUCCESS	The function completed without errors.
PCV_PLLNOTLOCKED	The EXM-14 was unable to lock onto a composite video signal

PCV_WRONGTYPE	The video input is programmed for RGB video. Use the **PCV_GetRGB** function to obtain the level and timing parameters for RGB video input.

PCV_GetContrast

void PCV_GetContrast(*fpContrast*)

This function returns the current contrast value. The contrast control the sharpness of the video display against the VGA display. This function is disabled on later versions of the hardware.

Parameter	Type	Description
fpContrast	**unsigned char far***	Contrast levels vary between 0 and 255.



PCV_GetRGB

int PCV_GetRGB(*fpLevel*)

This function returns the level and timing parameters for RGB video input.

Parameter	Type	Description
fpLevel	**unsigned long far***	Gain of the red, blue, and green D/A converters. Each gain is an 8-bit value that specifies a roughly linear range between 0 and 3 volts. The blue gain is specified in bits 7 through 0, the green gain is specified in bits 15 through 8, and the red gain is specified in bits 23 through 16. Bits 31 through 24 are reserved and should be zero.

The function returns a value for every parameter that is not a null pointer. If the parameter is a null pointer, then that parameter is not returned.

Return Value	Description
PCV_SUCCESS	The function completed without errors.

PCV_WRONGTYPE The video input is programmed for composite video. Use the **PCV_GetComposite** function to obtain the level and timing parameters for composite video input.

PCV_GetScrWinMap

void PCV_GetScrWinMap(*fpxFB, fpyFB, fpxS, fpyS, fpcx, fpcy*)

This function returns the screen window where the frame buffer will be displayed. The function returns the position of the screen window and of the frame buffer window, as well as the common size of the two windows.

Parameter	Type	Description
fpxFB	**unsigned short far***	X-position of the frame buffer window's upper left corner in frame buffer coordinates.
fpyFB	**unsigned short far***	Y-position of the frame buffer window's upper left corner in frame buffer coordinates.
fpxS	**unsigned short far***	X-position of the screen window's upper left corner in VGA screen coordinates.
fpyS	**unsigned short far***	Y-position of the screen window's upper left corner in VGA screen coordinates.
fpcx	**unsigned short far***	Width of the screen and frame buffer windows in their respective coordinates.
fpcy	**unsigned short far***	Height of the screen and frame buffer windows in their respective coordinates.

The function returns a value for every parameter that is not a null pointer. If the parameter is a null pointer, then that parameter is not returned.

PCV_GetState

void PCV_GetState(*fpState*)

This function copies the current PC video configuration to a local data structure.

Parameter	Type	Description
fpState	**PCV_t far***	Pointer to data structure in which to store the PC video state.

This functions saves the state variables set by **PCV_Display**, **PCV_Freeze**, **PCV_SetAcqWinMap**, **PCV_SetScrWinMap**, **PCV_SetComposite** and **PCV_SetRGB**. This function does not save the contents of the frame buffer.

The PC video state is not changed.

PCV_ReadRect

int PCV_ReadRect(*xFB, yFB, cx, cy, pBuf*)

This function copies a rectangular region of the frame buffer into local memory. Acquisition of the video input data is temporarily suspended while the function runs.

Parameter	Type	Description
xFB	**unsigned short**	X-position of the frame buffer rectangle's upper left corner in frame buffer coordinates.
yFB	**unsigned short**	Y-position of the frame buffer rectangle's upper left corner in frame buffer coordinates.
cx	**unsigned short**	Width of the frame buffer rectangle in frame buffer coordinates. The X-coordinate of the frame buffer rectangle's lower right corner is *xFB*+*cx*-1.
cy	**unsigned short**	Height of the frame buffer rectangle in frame buffer coordinates. The Y-coordinate of the frame buffer rectangle's lower right corner is *yFB*+*cy*-1.
pBuf	**unsigned short [***cy***][***cx***]**	Location of a two ORed array to store the 16-bit frame buffer data. Pixels on the same scan line are physically adjacent.

This function works directly with the frame buffer. The function does not work with the video input's window into the frame buffer, nor with the VGA screen's window into the frame buffer.

The function verifies that the frame buffer rectangle coordinates lie within the physical frame buffer. The implied size of the local buffer is not checked for validity.

Return Value	Description
PCV_SUCCESS	The function completed without errors.
PCV_FBBOUNDS	Either (*xFB*, *yFB*) or (*xFB*+*cx*-1, *yFB*+*cy*-1) is outside the physical boundaries of the frame buffer.

Beware that this function must be called multiple times to read the entire frame buffer because at most 64 KBytes may be transferred to any one buffer.

PCV_SetAcqWinMap

int PCV_SetAcqWinMap(*xVI*, *yVI*, *cxVI*, *cyVI*, *xFB*, *yFB*, *cxFB*, *cyFB*)

This function defines the video input region to be acquired. The function specifies the position and size of the video input window, as well as the position and size of the acquired image in the frame buffer.

Parameter	Type	Description
xVI	**unsigned short**	X-position of the video input window's upper left corner in video input coordinates.
yVI	**unsigned short**	Y-position of the video input window's upper left corner in video input coordinates.
cxVI	**unsigned short**	Width of the video input window in video input coordinates. The X-coordinate of the video input window's lower right corner is *xVI+cxVI-1*.
cyVI	**unsigned short**	Height of the video input window in video input coordinates. The Y-coordinate of the video input window's lower right corner is *yVI+cyVI-1*.

xFB	**unsigned short**	X-position of the frame buffer window's upper left corner in frame buffer coordinates.
yFB	**unsigned short**	Y-position of the frame buffer window's upper left corner in frame buffer coordinates.
cxFB	**unsigned short**	Width of the frame buffer window in frame buffer coordinates. The X-coordinate of the frame buffer window's lower right corner is *xFB*+*cxFB*-1.
cyFB	**unsigned short**	Height of the frame buffer window in frame buffer coordinates. The Y-coordinate of the frame buffer window's lower right corner is *yFB*+*cyFB*-1.

The video input window will be scaled down to approximately the specified size of the frame buffer window. The EXM-14 can support video source resolutions of up to 1024 x 512 pixels. The scaling is approximate because it is performed by the EXM-14 hardware dropping some number of video input horizontal pixels per 64 horizontal pixels, or by dropping some number of video input scan lines per 64 scan lines. The function always rounds the scale factor up so that the least number of rows and columns greater than *cxFB* and *cyFB* are acquired.

The function verifies that the frame buffer window coordinates lie within the physical frame buffer. The video input window coordinates are not checked for validity.

Return Value	**Description**
PCV_SUCCESS	The function completed without errors.
PCV_FBBOUNDS	Either (*xFB*, *yFB*) or (*xFB*+*cxFB*-1, *yFB*+*cyFB*-1) is outside the physical boundaries of the frame buffer.
PCV_BADSCALE	An illegal scale factor between the video input window and the frame buffer was specified. This can occur if either *cxFB* or *cyFB* is zero. It can also occur if either*cxFB*/*cxVI* or *cyFB*/*cyVI* is greater than one.

PCV_SetColorKey

void PCV_SetColorKey(*key*)

This function sets the current color key. The color key controls which pixels inside the VGA screen window display the frame buffer, and which pixels display VGA screen data.

Parameter	Type	Description
key	**unsigned short**	The color key for the VGA screen window. See the description below for what values the color key can take on.

The color key is constructed from two eight bit values: a data mask in the upper eight bits and a data value in the lower eight bits. A '1' bit in the data mask causes the corresponding bit in the data value not to participate in the comparison. Normally, the upper eight bits are '0' so an exact color match is required to display the frame buffer. The cases where all eight mask bits are '1' are special cases that are handled specially.

Color Key Value	Description
0xFF00	A color key of 0xFF00 means that all pixels inside the VGA screen window will be replaced with the frame buffer data.
0xFFFF	A color key of 0xFFFF means that all pixels inside the VGA screen window will display the VGA screen data.
0x00*HH*	A color key in the range 0x0000 to 0x00FF means that those pixels inside the VGA screen window with a VGA data value equal to the low order eight bits *HH* will be replaced with the frame buffer data, and all other pixel values will display the VGA screen data.
0x*MMHH*	A color key with any but not all of the upper eight bits *MM* nonzero means that those pixels inside the VGA screen window with VGA data value equal to the low order eight bits *HH* when masked by *MM*. In this case, each upper byte bit acts as a mask, and suppresses the compare on those data bits for which a '1' is written to the corresponding lower byte bit position *HH*.

PCV_SetComposite

int PCV_SetComposite(*Type*)

This function sets up the level and timing parameters for composite video input. The function programs the EXM-14 for composite video signals, automatically detects the composite video type, and initializes the timing and level parameters accordingly.

Parameter	Type	Description
Type	**int**	The type of the composite video is explicitly specified. The type may optionally be combined with **PCV_SVHS**, which directs the function to adjust the timing and gain levels slightly for Super VHS video signals.

There are three types of composite video signals explicitly supported.

Type	Description
PCV_NTSC	NTSC
PCV_PAL	PAL
PCV_SECAM	SECAM

The composite hue and saturation levels are always set to their nominal midpoint values. Use the **PCV_SetCompositeLevel** function to adjust these.

Return Value	Description
PCV_WRONGTYPE	Invalid composite type specified.
PCV_SUCCESS	The function completed without errors.

Use the **PCV_SetRGB** function to set up the EXM-14 for RGB video input.

PCV_SetCompositeLevel

void PCV_SetCompositeLevel(*Hue*, *Saturation*)

This function adjusts the level parameters for composite video input.

Parameter	Type	Description
Hue	**unsigned char**	Hue levels vary between 0 and 255.
Saturation	**unsigned char**	Saturation levels vary between 0 and 255.

Use the **PCV_SetComposite** function to set the composite video timing.

PCV_SetContrast

void PCV_SetContrast(*contrast*)

This function ORed the contrast between the video and VGA display. This function is disabled on later versions of the hardware.

Parameter	Type	Description
contrast	**unsigned char**	Value that is programmed into the DAC which controls the contrast.

PCV_SetRGB

void PCV_SetRGB(*Level*)

This function sets up the level and timing parameters for RGB video input. The function programs the EXM-14 for RGB video signals, and initializes the timing and level parameters to the specified levels.

Parameter	Type	Description
Level	**unsigned long**	Gain of the red, blue, and green D/A converters. Each gain is an 8-bit value that specifies a roughly linear range between 0 and 3 volts. The blue gain is specified in bits 7 through 0, the green gain is specified in bits 15 through 8, and the red gain is specified in bits 23 through 16. Bits 31 through 24 are reserved and should be zero.

There are a few standards for RGB levels and timing. For example RS170 has NTSC timing with standardized 1V gain levels. The following manifest constants are defined for programming convenience.

Constant	Description
PCV_RS170LEVEL	Gain levels for RS-170 (1 volt).
PCV_RS343LEVEL	Gain levels for RS-343 (0.714 volt).

This function does not wait to determine if the phase locked loop has found the RGB video signal.

Use the **PCV_SetComposite** function to set up the EXM-14 for composite video input.

PCV_SetScrWinMap

int PCV_SetScrWinMap(*xFB*, *yFB*, *xS*, *yS*, *cx*, *cy*, *fFit*)

This function defines the screen window where the frame buffer will be displayed. The function specifies the position of the screen window and of the frame buffer window, as well as the common size of the two windows.

Parameter	Type	Description
xFB	**unsigned short**	X-position of the frame buffer window's upper left corner in frame buffer coordinates.
yFB	**unsigned short**	Y-position of the frame buffer window's upper left corner in frame buffer coordinates.
xS	**unsigned short**	X-position of the screen window's upper left corner in VGA screen coordinates.
yS	**unsigned short**	Y-position of the screen window's upper left corner in VGA screen coordinates.
cx	**unsigned short**	Width of the screen and frame buffer windows in their respective coordinates. The X-coordinate of the frame buffer window's lower right corner is *xFB*+*cx*-1. The X-coordinate of the screen window's lower right corner is *xS*+*cx*-1.

cy	**unsigned short**	Height of the screen and frame buffer windows in their respective coordinates. The Y-coordinate of the frame buffer window's lower right corner is *yFB*+*cy*-1. The Y-coordinate of the screen window's lower right corner is *yS*+*cy*-1.
fFit	**unsigned short**	If fFit is zero, then no scaling is done to fit the video display into the screen window. If fFit is non zero, then the video display is scaled to fit into the screen window.

The function verifies that the frame buffer window coordinates lie within the physical frame buffer. The screen window coordinates are not checked for validity.

Return Value	Description
PCV_SUCCESS	The function completed without errors.
PCV_FBBOUNDS	Either (*xFB*, *yFB*) or (*xFB*+*cx*-1, *yFB*+*cy*-1) is outside the physical boundaries of the frame buffer.

PCV_SetState

void PCV_SetState(*pState*)

This function sets the PC video state to a particular configuration.

Parameter	Type	Description
pState	**PCV_t ***	Pointer to data structure containing the PC video state previously saved by **PCV_GetState**.

This functions restores the state variables set by **PCV_Display**, **PCV_Freeze**, **PCV_SetAcqWinMap**, **PCV_SetScrWinMap**, **PCV_SetComposite** and **PCV_SetRGB**. This function does not change the contents of the frame buffer.

No checking of the internal format is done.

PCV_WriteRect

int PCV_WriteRect(*xFB*, *yFB*, *cx*, *cy*, *pBuf*)

This function copies local memory into a rectangular region of the frame buffer. Acquisition of the video input data is temporarily suspended while the function runs.

Parameter	Type	Description
xFB	**unsigned short**	X-position of the frame buffer rectangle's upper left corner in frame buffer coordinates.
yFB	**unsigned short**	Y-position of the frame buffer rectangle's upper left corner in frame buffer coordinates.
cx	**unsigned short**	Width of the frame buffer rectangle in frame buffer coordinates. The X-coordinate of the frame buffer rectangle's lower right corner is *xFB*+*cx*-1..
cy	**unsigned short**	Height of the frame buffer rectangle in frame buffer coordinates. The Y-coordinate of the frame buffer rectangle's lower right corner is *yFB*+*cy*-1.
pBuf	**unsigned short [***cy***][***cx***]**	Location of a two ORed array that contains the 16-bit frame buffer data. Pixels on the same scan line are physically adjacent.

This function works directly with the frame buffer. The function does not work with the video input's window into the frame buffer, nor with the VGA screen's window into the frame buffer.

The function verifies that the frame buffer rectangle coordinates lie within the physical frame buffer. The implied size of the local buffer is not checked for validity.

Return Value	Description
PCV_SUCCESS	The function completed without errors.
PCV_FBBOUNDS	Either (*xFB*, *yFB*) or (*xFB*+*cx*-1, *yFB*+*cy*-1) is outside the physical boundaries of the frame buffer.

Beware that this function must be called multiple times to write the entire frame buffer because at most 64 KBytes may be transferred to any one buffer.

Supported Environments

This section describes the runtime and development environments supported by the PC Video software.

Windows

The PC Video software runs under Microsoft Windows 3.x in the standard and enhanced protected modes. The PC Video software is not supported when running Windows in real mode.

The Windows distribution ORed the following files:

File	Description
PCVIDEO.LIB	A Windows library that contains all the functions support by the EXM-14 software.
PCVIDEO.DLL	A Windows dynamic link library that implements the EXM-14 device driver. You can link your application with an IMPORTS PCVIDEO.DLL in your DEF file or specific PCVIDEO.LIB as one of the libraries that your application depends on.
PCVIDEO.H	The C include file that defines the function prototypes, the manifest constants, and the data structures used by PCVIDEO.DLL.
PCVTUNER.EXE	An interactive Windows application that allows the user to manipulate and adjust most of the features available through the PC Video software library.



SAMPLES\DEMO.C	Source code for a demonstration Windows application that illustrates how to connect the PC Video software to Windows. The program set up the acquisition and display windows. It demonstrates how to handle a window when the size is changed or the window is moved. The freeze frame and scaling functions are also tested. The application can also decode all the different video sources which the board can accept.
SAMPLES\DEMO.H	The C include file that defines the structures and variables used in DEMO.C
SAMPLES\DEMO.RC	The windows resource file that describes the menu bar and dialog boxes found in the demo program.
SAMPLES\DEMO.DEF	The module definition file for the demo program.
SAMPLES\DEMO.MAK	The Microsoft make file for the demo program.

6 6

OS/2

The PC Video software is a Windows application that runs under Microsoft OS/2 1.21, 1.3, and 2.x with Presentation Manager. The PC Video software is not supported when Presentation Manager is not loaded.

The OS/2 distribution ORed the following files:

File	Description
PCVIDEO.LIB	An OS/2 library which contains all the functions supported by the EXM-14 software.

PCVIDEO.DLL	An OS/2 dynamic link library that implements the non-privileged portion of the EXM-14 device driver. You can link your application with an IMPORTS PCVIDEO.DLL in your DEF file or specify PCVIDEO.LIB as one of the libraries that your application depends on.
PCVIDEO.SYS	An OS/2 loadable device driver that implements the privileged portion of the EXM-14 device driver. This driver is used by PCVIDEO.DLL, and must be present when PCVIDEO.DLL is loaded.
PCVIDEO.H	The C include file that defines the function prototypes, the manifest constants, and the data structures used by PCVIDEO.DLL.
PCVTUNER.EXE	An interactive OS/2 PM application that allows the user to manipulate and adjust most of the features available through the PC Video software library. This program uses the Microsoft WLO DLL to run as a Windows application under OS/2 PM.
SAMPLES\PMDEMO.C	Source code for a demonstration PM application that illustrates how to connect the PC Video software. It set up the acquisition and display window. It can also handle the window being resized or moved.
SAMPLES\PMDEMO.DEF	The module definition file for demo program.
SAMPLES\PMDEMO.MAK	A Microsoft make file for the demo.
WLO\DLL*.*	WLO mapping layer dynamic_link libraries which are necessary for the pcvideo library.

NOTES

6 6

7. Support and Service

In North America

Technical Support

RadiSys maintains a technical support phone line at (503) 646-1800 that is staffed weekdays (except holidays) between 8 AM and 5 PM Pacific time. If you have a problem outside these hours, you can leave a message on voice-mail using the same phone number. You can also request help via electronic mail or by FAX addressed to RadiSys Technical Support. The RadiSys FAX number is (503) 646-1850. The RadiSys E-mail address on the Internet is *support@radisys.com*. If you are sending E-mail or a FAX, please include information on both the hardware and software being used and a detailed description of the problem, specifically how the problem can be reproduced. We will respond by E-mail, phone or FAX by the next business day.

Technical Support Services are designed for customers who have purchased their products from RadiSys or a sales representative. If your RadiSys product is part of a piece of OEM equipment, or was integrated by someone else as part of a system, support will be better provided by the OEM or system vendor that did the integration and understands the final product and environment.

Bulletin Board

RadiSys operates an electronic bulletin board (BBS) 24 hours per day to provide access to the latest drivers, software updates and other information. The bulletin board is not monitored regularly, so if you need a fast response please use the telephone or FAX numbers listed above.

The BBS operates at up to 14400 baud. Connect using standard settings of eight data bits, no parity, and one stop bit (8, N, 1). The telephone number is (503) 646-8290.

Repair Services

Factory Repair Service is provided for all RadiSys products. Standard service for all RadiSys products covers factory repair with customers paying shipping to the factory and RadiSys paying for return shipment. Overnight return shipment is available at customer expense. Normal turn-around time for repair and re-certification is five working days.

Quick Exchange services (immediate shipment of a loaner unit while the failed product is being repaired) or other extra-cost services can be arranged, but need to be negotiated in advance to allow RadiSys to pool the correct product configurations. RadiSys does not maintain a general "loaner" pool: units are available only for customers that have negotiated this service in advance.

RadiSys does not provide a fixed-price "swap-out" repair service, as customers have indicated that issues of serial number tracking and version control make it more convenient to receive their original products back after repair.

Warranty Repairs

Products under warranty (see warranty information in the front of this manual) will have manufacturing defects repaired at no charge. Products sent in for warranty repair that have no faults will be subject to a recertification charge. Extended Warranties are available and can be purchased at a standard price for any product still under warranty. RadiSys will gladly quote prices for Extended Warranties on products whose warranties have lapsed; contact the factory if this applies.

Customer induced damage (resulting from misuse, abuse, or exceeding the product specifications) is not covered by the standard product warranty.

Non-Warranty Services

There are several classes of non-warranty service. These include repair of customer induced problems, repairs of failures for products outside the warranty period, recertification (functional testing) of a product either in or out of warranty, and procurement of spare parts.

All non-warranty repairs are subject to service charges. RadiSys has determined that pricing repairs based on time and materials is more cost-effective for the customer than a flat-rate repair charge. When product is received, it will be analyzed and, if appropriate, a cost estimate will be communicated to the customer for authorization. After the customer authorizes the repair and billing arrangements have been made, the product will be repaired and returned to the customer.

A recertification service is provided for products either in or out of warranty. This service will verify correct operation of a product by inspection and testing of the product with standard manufacturing tests. There is a product-dependent charge for recertification.

There are only a few components that are generally considered field-repairable, but, because RadiSys understands that some customers want or need the option of repairing their own equipment, all components are available in a spares program. There is a minimum billing charge associated with this program.

Arranging Service

To schedule service for a product, please call RadiSys Technical Support directly at (503) 646-1800. Have the product model and serial numbers available, along with a description of the problem. A Technical Support representative will issue a Returned Materials Authorization (RMA) number, a code number by which we track the product while it is being processed. Once you have received the RMA number, follow the instructions of the Technical Support representative and return the product to us, freight prepaid, with the RMA number clearly marked on the exterior of the package. If possible re-use the original shipping containers and packaging. In any case, be sure you follow good ESD-control practices when handling the product, and ensure that anti-static bags and packing materials with adequate padding and shock-absorbing properties are used.

Ship the product, freight prepaid, to the following address:

Product Service Center
RadiSys Corporation
15025 SW Koll Parkway
Beaverton, Oregon 97006-6902

When shipping the product, include the following information: return address, contact names and phone numbers in purchasing and engineering, and a description of the suspected problem. Any ancillary information that might be helpful with the debugging process will be appreciated.

Other Countries

Contact the sales organization from which you purchased your RadiSys product for service and support.

Index

82C9001A, 19

A

Acquisition and Display Process, 20
Altitude, 2
Analog Input Control, 14
Automatic gain control, 15, 18
Auxiliary connector, 9

B

BLANK, 10
BLUE (live video), 10
BLUE (VGA), 10
Brooktree triple video DAC, 19

C

Card enable, 13
CDEN, 13
CGA, 7
Chips & Technologies PC-video chip, 19
color keying, 17
Color space converter, 18
Color-keying, 19
COMP/SYNC, 11
Composite Decoder Parameters, 15
Composite video, 15, 17, 27
Configuration, 27
Configuration Registers, 13
Connecting a Monitor, 7
Connectors, 9
Coordinate systems, 23
Current, 2

D

DCLK, 11
DCSC Registers, 16
Digital Color Space Converter, 16
DMSD, 16, 18
Dot Clock, 19

E

EGA, 7
Environmental specifications, 2
External video source, 6

F

FB space, 24
Feature connector, 19
FIELD signal, 18
Frame buffer, 17, 21, 22
Frame buffer coordinate space, 23

G

General Purpose I/O Control register, 14
GREEN (live video), 11
GREEN (VGA), 11

H

Horizontal scaling, 21
Horizontal Sync, 19
HSYNC, 11
Humidity, 2

I I

I

I/O Base, 13
I2C bus, 14

K

Keying, 19

L

Luminance/chrominance, 16

M

Microsoft OS/2, 49
Microsoft Windows, 1, 47
Monitor, 7
Multifrequency monitor, 7
Multiscan monitor, 7
Multisync monitor, 7

N

NTSC, 16, 18, 20, 21

O

OS/2, 49
OS/2 Presentation Manager, 1

P

P0-P7, 11
PAL, 16, 18, 21
PCV_Display, 31
PCV_FillRect, 32
PCV_Freeze, 33
PCV_GetAcqWinMap, 33
PCV_GetColorKey, 34
PCV_GetComposite, 35
PCV_GetContrast, 36
PCV_GetRGB, 36
PCV_GetScrWinMap, 37
PCV_GetState, 37
PCV_ReadRect, 38
PCV_SetAcqWinMap, 26, 39
PCV_SetColorKey, 41
PCV_SetComposite, 42
PCV_SetCompositeLevel, 43
PCV_SetContrast, 43
PCV_SetRGB, 43
PCV_SetScrWinMap, 25, 44
PCV_SetState, 45
PCV_WriteRect, 46
PCVIDEO dynamic link library, 27
PCVIDEO.DLL, 48, 49
PCVIDEO.H, 48, 49
PCVIDEO.LIB, 47, 49
PCVIDEO.SYS, 49
PCVTUNER, 28
PCVTUNER.EXE, 48, 49
Philips SAA7191 multistandard decoder, 18
Power, 2
Presentation Manager, 49

R

RED (live video), 11
RED (VGA), 11
Resolution modes, 7
RGB, 14, 17, 20, 21
RS-170, 21
RS-170/RS-343, 17
RS-170A, 18
RS-343, 21

S

S space, 24
SAA7191 DMSD, 14
SAA7192, 16, 18
SAA7192 DCSC, 14
SAMPLES\\DEMO.C, 48
SAMPLES\\DEMO.DEF, 48
SAMPLES\\DEMO.H, 48
SAMPLES\\DEMO.MAK, 48
SAMPLES\\DEMO.RC, 48
SAMPLES\\PMDEMO.C, 49
SAMPLES\\PMDEMO.DEF, 50

SAMPLES\\PMDEMO.MAK, 50
Screen coordinate space, 23
SECAM, 16, 18, 21
Shock, 2
Signetics TDA8444 A-D converter, 17
Slot register, 14
Super VHS, 18
Supported Environments, 47
SVHS, 15, 16, 18

T

TDA8444, 16, 19
TDA8444 DAC, 14
Temperature, 2

V

Vertical scaling, 21
Vertical Sync, 19
VGA, 7
VGA Connector, 9
VI space, 24
Vibration, 2
Video Input, 21
Video input coordinate space, 23
Video Input Source, 6
Video level information, 21
Video output, 22
Video Output Control, 16
VSYNC, 11

W

Weight, 2

X

X-Y coordinates, 17

Y

Y/C format., 16

NOTES

Appendix A: Video Standards

Timing parameters summary

RS-170, M/PAL
Horizontal line frequency: 15,750 Hz (period: 63.492 uS)
Horizontal blanking interval: 10.80 uS
Time from HS to active video: 9.6 uS
Field frequency: 60 Hz (period: 16.667 mS)
Lines per frame: 525

RS-170A, M/NTSC, (US, Japan)
Horizontal line frequency: 15,734 Hz (period: 63.556 uS)
Horizontal blanking interval: 10.90 uS
Time from HS to active video: 9.4 uS
Field frequency: 59.94 Hz (period: 16.6833 mS)
Lines per frame: 525

N/PAL
Horizontal line frequency: 15,625 Hz (period: 64.00 uS)
Horizontal blanking interval: 10.88 uS
Time from HS to active video: 9.6 uS
Field frequency: 50 Hz (period: 20 mS)
Lines per frame: 625

B, G, H, I, D, K, K1, L/PAL or SECAM
Horizontal line frequency: 15,625 Hz (period: 64.00 uS)
Horizontal blanking interval: 12.00 uS
Time from HS to active video: 10.50 uS
Field frequency: 50 Hz (period: 20 mS)
Lines per frame: 625

Level summary

RS-170 with sync
Absolute amplitude: 1.4V P-P.
Blanking level: 0 IRE (0.40 volts)
Black level: 7.5 IRE (0.475 volts)
Reference white: 100 IRE (1.40 volts)
Sync level: -40 IRE (0 volts)

RS-170 without sync
Absolute amplitude: 1.0V P-P.
Blanking level: 0 IRE (0 volts)
Black level: 7.5 IRE (0.075 volts)
Reference white: 100 IRE (1.00 volts)

RS-170A
(same as RS-170 with sync)

RS-343 with sync
Absolute amplitude: 1.00V P-P.
Blanking level: 0 IRE (0.286 volts)
Black level: 7.5 IRE (0.339 volts)
Reference white: 100 IRE (1.00 volts)
Sync level: -40 IRE (0 volts)

RS-343 without sync
Absolute amplitude: 0.714V P-P.
Blanking level: 0 IRE (0 volts)
Black level: 7.5 IRE (0.053 volts)
Reference white: 100 IRE (0.714 volts)

M/NTSC, M/PAL, N/PAL
(Typical level 1V. Absolute level not defined by spec.)
(see RS-170 with sync)

B, G, H, I, D, K, K1, L/PAL or SECAM
(Typical level 1V. Absolute level not defined by spec.)
Blanking level: 0 IRE (0.300 volts)
Black level: 0 IRE (0.300 volts)
Reference white: 100 IRE (1.00 volts)
Sync level: -43 IRE (0 volts)

Appendix B: Glossary

B B

ADC or *A/D* - Analog-to-Digital Converter. Converts a voltage to a digital binary number.

Blanking level - "blacker than black", a level guaranteed to blank the display. (PAL and SECAM define reference black as the same as blanking level).

Chrominance or *Chroma* - the color information in a video signal.

Composite - Video signal where sync, color (chroma) and intensity (luma) are encoded in one waveform.

DAC - Digital-to-Analog converter. Converts a digital value to a voltage

IRE - A numerical representation of video waveforms. Unitless. For NTSC, RS170, and RS343, 100 IRE is defined as reference white, 0 blanking level. Reference black is 7.5 IRE. Sync is -40 below blanking. PAL and SECAM define these points slightly differently.

Luminance or *Luma* - the intensity information in a video signal.

NTSC - A composite video encoding standard used in north America and Japan. There are many variations, but the generic NTSC/M is the most common.

PAL - A composite video standard used in Europe. Again, there are several variations. PAL/B and PAL/G are the most common.

Reference black - the "blackest" black level

Reference white - the "whitest" white level

RGB - Color video that is provided on 3 separate signals for Red, green, and blue.

B B

RS-170 - a standard for video specifying timing and levels. Commonly used for TV closed circuit and studio facilities.

RS-343 - a higher-resolution video graphics standard. Defines levels differently than RS-170. Allows variation in timing. VGA monitors use RS-343 video.

Sync - the timing information for the video waveform. May be contained on a composite video signal or separately.

Sync-on-green - RGB video format where the sync signal is carried similar to composite video on the Green video signal.

SVHS - Super VHS. A form of composite video where chroma is carried on a separate signal from the luma and sync.

SECAM - Another standard, used in France and some third-world countries. SECAM differs from PAL and NTSC in that the color difference signal is frequency-modulated instead of QAM (quadrature amplitude modulated).

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP