

# VAC SDK Software Development Kit

for VAC104plus, VAC2000 and VACPCI



The VAC SDK library is an extensive collection of demonstration applications, library routines and documentation for the Advanced Micro Peripherals VAC104plus, VAC2000 and VACPCI Video Annotation Controllers.



The SDK and is designed to run on Microsoft Windows, Linux and QNX. All the demonstration applications function with a PAL or NTSC video source and drive an NTSC, PAL, VGA or TV display.

The VAC SDK for Windows is written in Microsoft Visual C++ 6. It contains demonstration applications that utilise Microsoft's Foundation Class Library (MFC) and Graphics Device Interface (GDI) to provide the graphical and text functions.

Extensive  
collection  
of elegant  
routines and  
demonstration  
applications

**Advanced Micro Peripherals Ltd**  
Cambridge, CB6 2HY, England  
Tel (+44) 1353 659500  
Fax (+44) 1353 659600  
[sales@ampltd.com](mailto:sales@ampltd.com)  
<http://www.ampltd.com>

**Advanced Micro Peripherals Inc.**  
New York, NY, USA, 10001  
Tel (+1) 212 951 7205  
Fax (+1) 212 951 7205  
[sales@amp-usa.com](mailto:sales@amp-usa.com)  
<http://www.amp-usa.com>



# VAC SDK Software Development Kit

for VAC104plus, VAC2000 and VACPCI



**Advanced Micro  
Peripherals**

THE EMBEDDED VIDEO EXPERTS

The MFC interfaces with the GDI and provides classes that can be used in building application programs to run on Windows. Classes are provided for all graphical user interface elements, for example windows, frames, menus, tool bars and status bars.

The GDI provides the core graphics functions. It controls the display of text and fonts, draws lines and curves and provides scaling functions.

The VAC SDK for Linux contains demonstration applications that can be run directly on Linux. A version is also available which interfaces with any X Window Manager.

The VAC SDK for QNX contains demonstration applications which interface with QNX Photon and use the standard Photon drawing functions.

The demonstration version of the SDK library is fully featured. However, certain functions will not work after one hour of continuous use of the library. All function calls are identical between the demonstration and full version so that applications developed with the demonstration version will continue to work with the full version after recompilation.

**Complete,  
comprehensible  
documentation**

**Advanced Micro Peripherals Ltd**  
Cambridge, CB6 2HY, England  
Tel (+44) 1353 659500  
Fax (+44) 1353 659600  
[sales@ampltd.com](mailto:sales@ampltd.com)  
<http://www.ampltd.com>

**Advanced Micro Peripherals Inc.**  
New York, NY, USA, 10001  
Tel (+1) 212 951 7205  
Fax (+1) 212 951 7205  
[sales@amp-usa.com](mailto:sales@amp-usa.com)  
<http://www.amp-usa.com>

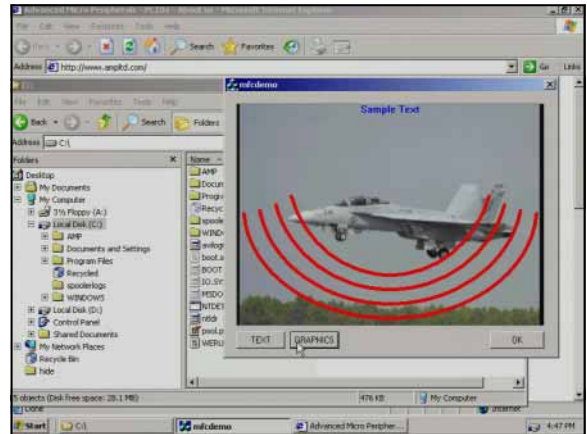


# VAC SDK Software Development Kit

for VAC104plus, VAC2000 and VACPCI

## **mfcdemo**

The mfcdemo application provides an example of using Microsoft's MFC library. This application uses MFC to draw text and simple graphics overlaid on live video and provide basic interaction with the user.



**mfcdemo**

## **windemo**

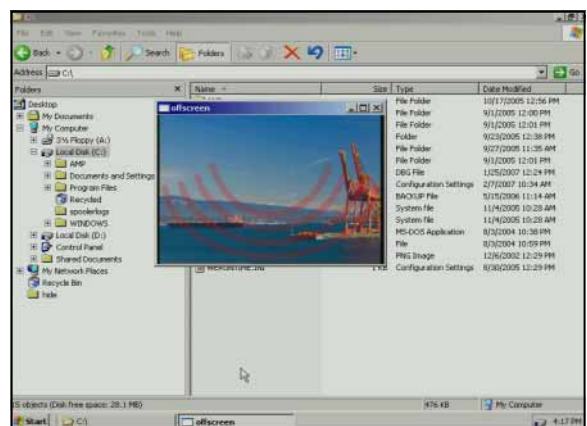
The windemo application demonstrates the manipulation of a window displaying live video and graphics. The window can be resized and moved about the display. The window can be displayed full-screen with moving graphics alpha-blended with the live video. This application makes use of both GDI functions and SDK functions.



**windemo**

## **offscreen**

The offscreen application gives an example of using offscreen memory to combine live video and graphics. GDI functions are used to draw onto a bitmap which has been created using a device context. The bitmap is then copied to the display memory. The alpha-blending level between the live video and graphics can be varied by the user.



**offscreen**

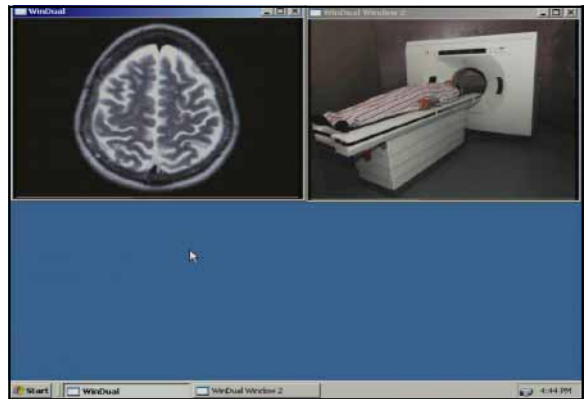


# VAC SDK Software Development Kit

for VAC104plus, VAC2000 and VACPCI

## Windual (VAC104plus, VAC2000 only)

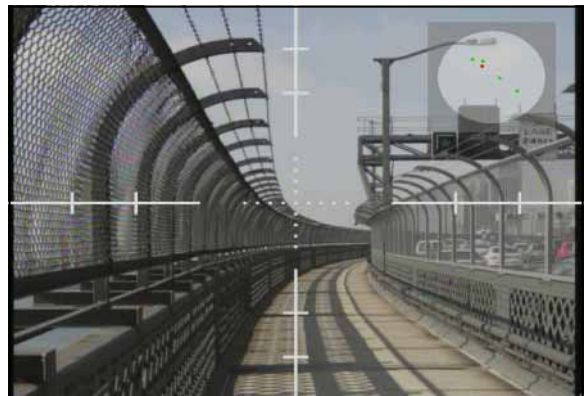
The windual application demonstrates the manipulation of two independent live video windows. The windows can be moved about the display, mimimised and restored. Either window can be moved to overlay the other and the two windows can be alpha-blended together using varying levels of alpha-blending.



**Windual**

## ReticuleRadar

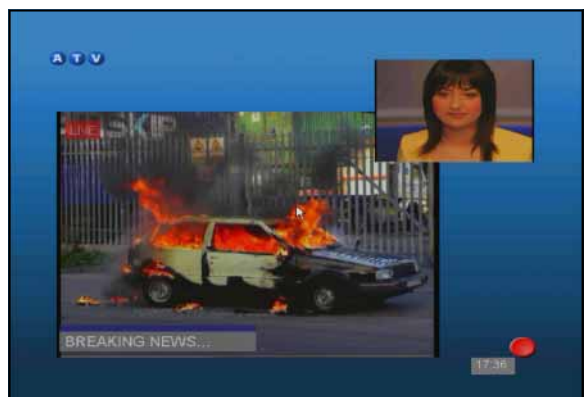
The ReticuleRadar application displays a single video input full screen. A reticule is drawn using GDI drawing functions and overlaid onto the video. A simple radar display tracking five objects is also alpha blended with the video.



**ReticuleRadar**

## DualWindow (VAC104plus, VAC2000 only)

The DualWindow demonstration application sets up two video windows of differing sizes. Text drawn by the GDI is overlaid onto the video and alpha-blended using two alpha rectangles.



**DualWindow**



# VAC SDK Software Development Kit

for VAC104plus, VAC2000 and VACPCI

## OffscreenAlphaText

The OffscreenAlphaText demonstration application provides an example of the use of per-pixel alpha-blending and off-screen memory. Text is drawn onto background rectangles and stored in an offscreen memory buffer. This is then alpha-blended with the video such that the text is opaque and the rectangles are translucent. The result is displayed full-screen.



**OffscreenAlphaTest**

## SimpleAlphaText

The SimpleAlphaText application provides the same functions as OffscreenAlphaText except that it draws directly to the display instead of using offscreen memory buffers. Drawing to the display means that updates to the text or graphics may cause a flicker.



# VAC SDK Software Development Kit

for VAC104plus, VAC2000 and VACPCI

## General functions

InitLibrary	Initialise the library
InitHardware	Initialise the hardware interface
AMP_Init	Initialise the hardware
AMP_DeInit	De-initialise the hardware interface
SetVMode	Set the video output mode
GetHRes	Return the horizontal resolution in pixels
GetVRes	Return the vertical resolution in pixels
GetCurrentBpp	Return the number of bytes per pixel for the onscreen display
GetScreenOffset	Return the offset to the start of the visible display
GetVersionString	Return information about the current version of the library
WaitForVR	Wait for vertical retrace
GetMonitor	Return the virtual screen co-ordinates of the VAC104plus in multi-monitor situations.
SetClock	Set the memory clock to the specified value

## Capture and Overlay Functions

ProgramOutputViewPortWindow	Set the output viewport window parameters
SetInputViewPortWindow	Set the input viewport. Allows portions of the video window to be displayed and enlarged
SetOutputViewPortFunctionControl	Specify whether colour keying should be used
SetColorKey	Specify the color key
SetInputNormal	Sets the default input video standard
Freeze	Start and stop the video capture
StopCapture	Stop the Capture but leave the last captured frame visible
StartCapture	Restart the capture
ShowVacWindow	Enable the display of the overlay window
HideVacWindow	Disable the display of the overlay window
SetFrontWindow	Select the top window when two overlay windows overlap



# VAC SDK Software Development Kit

for VAC104plus, VAC2000 and VACPCI

SetContrast	Set the contrast for the captured video
SetSaturation	Set the saturation for the captured video
SetBrightness	Set the brightness for the captured video
SetHue	Set the hue for the captured video
SetVideoInput	Select the video input to the specified decoder
InitOverlay	Initialise the overlay window
MirrorX	Enable/disable mirroring of captured video
SetOverlayWindow	Set the source and output viewports
SetOverlayFormat	Set the source data format for the overlay window
GetOverlayLocation	Return the offset from the start of video memory of the overlay window
AddVRTimeoutCallback	Define the function to be called upon removal of the source video

## Drawing Functions

Under Windows the standard Windows drawing primitives are the preferred mechanism for drawing. The SDK provides the following additional functions:

SetVacPixel	Draw a pixel at the specified position
GetVacCapPixel	Return the colour of the specified pixel from a capture window
ClearScreen	Clears the screen to the specified color
RectFill	Fill a solid rectangle with the specified color
Line	Draw a single-pixel width line
VacEllipse	Draw the outline of an ellipse of specified size and position
VacEllipseFilled	Draw a filled ellipse of specified size and position
VacCircle	Draw the outline of a circle of specified size and position
VacCircleFilled	Draw a filled circle of specified size and position
VacArc	Draw the outline of a single quadrant of an ellipse
VacTriangle	Draw the outline of a triangle of specified size and position
VacTriangleFilled	Draw a filled triangle of specified size and position



# VAC SDK Software Development Kit

for VAC104plus, VAC2000 and VACPCI

VacPoly	Draw the outline of a polygon
VacPolyfilled	Draw a filled polygon
<b>Alpha-Blending Functions</b>	
EnableAlphaBlending	Enable/disable alpha-blending
SetAlphaValue	Set the alpha value for the graphics on top of video
SetAlphaMode	Set the alpha blending mode
SetAlphaTargets	Select the target for alpha-blending
SetAlphaWindow	Set the position and size of the alpha window
SetPixelAlpha	Set the alpha value for the specified pixel
ClearAlpha	Set all alpha values to specified value
Convertto8	Prepare data for blitting
RectAlpha	Set the alpha values within the outline of a rectangle to the specified value
RectAlphaOffset	Set the alpha values within a filled rectangle to the specified value
LineAlpha	Set the alpha values within a line of any dimensions to the specified value
VacEllipseAlpha	Set the alpha values within the outline of an ellipse to the specified value
VacEllipsefilledAlpha	Set the alpha values within a filled ellipse to the specified value
VacCircleAlpha	Set the alpha values within the outline of a circle to the specified value
VacCircleFilledAlpha	Set the alpha values within a filled circle to the specified value
VacArcAlpha	Set the alpha values within the outline of a single quadrant of an ellipse to the specified value
VacTriangleAlpha	Set the alpha values within the outline of a triangle to the specified value
VacTriangleFilledAlpha	Set the alpha values within a triangle to the specified value



# VAC SDK Software Development Kit

for VAC104plus, VAC2000 and VACPCI

VacPolyAlpha	Set the alpha values within the outline of a polygon to the specified value
VacPolyfilledAlpha	Set the alpha values within a polygon to the specified value
DrawStringOnAlpha	Write text to the alpha mask

## TV Functions

TvOutput	Enable and disable TV output
AdjustTVHorz	Adjust the horizontal position of the TV picture
AdjustTVVert	Adjust the vertical position of the TV Picture
FpOutput	Enable and disable the flat panel output

## Blitting Functions

SysBlit	Copy from system memory to the visible display
AcceleratedSysBlit	Copy from system memory to visible display or offscreen video memory
SysBlitAlpha	Blit data into alpha surface memory
AcceleratedSrcCopyBlt	Blit data from video memory to an offset in video memory
AcceleratedSrcCopyBlt2	Blit data from video memory to an offset in video memory (alternative parameters)
AcceleratedSrcCopyBlt3	Blit data from video memory to an offset in video memory (alternative parameters)
AcceleratedSrcCopyBlt4	Copy data from small sections of memory areas
GetMemLocation	Return the location of free video memory



for VAC104plus, VAC2000 and VACPCI

## Text Functions

Under windows the standard text functions are the preferred method for drawing text.

SetFont	Select the current font array
PrintString	Print the specified string at the current cursor location
MoveCursX	Move the cursor to the specified x co-ordinate
MoveCursY	Move the cursor to the specified y co-ordinate
DrawString	Draw a text string using the specified foreground and background colours
DrawStringOffset	Draw a text string using the specified foreground and background colours at a memory offset location.
DrawChar	Draw a character using the specified foreground and background colours.
GetVacTextExtent	Return the width and height of the specified text string
GetVacTextMetrics	Obtain the metric data of the current font

