

# Software Development Kit

## VSP2000 12-Channel Video Surveillance Processor



The VSP2000 SDK is an extensive collection of demonstration applications, library routines and instructions for utilising the Advanced Micro Peripherals VSP2000 12-channel PC/104plus Video Surveillance Processor.



The SDK provides complete control of both the monitor and record paths of the VSP2000 and a full DirectShow interface. The applications function with both PAL and NTSC video input.

The SDK contains a Windows Driver Model (WDM) kernel streaming driver and plugin filters which allow the VSP2000 custom interfaces to be accessed directly from within DirectShow. DirectShow graphs can be built using the VSP2000 as the video data source. The plugin filters expose custom VSP2000 DirectShow interfaces for controlling the advanced features of the VSP2000.

Custom WDM  
interface with  
complete control  
of display,  
record and  
motion-detection  
functions

# Software Development Kit

## VSP2000 12-Channel Video Surveillance Processor

### Demonstration Applications

The VSP2000 SDK Demonstration Applications build the required filtergraph using standard DirectShow techniques. When live video is required the DirectShow video renderer receives the video and overlays it on the display. When a still image is required for display or for saving as a bitmap, the DirectShow SampleGrabber filter is used.

The preview performance of the system is dependent on the capabilities of the display adapter. Most modern display devices should be capable of providing the overlay support required by the application.

### Vspdemo

The vspdemo application is intended to be the initial test application to confirm that the VSP2000 is working correctly.



**Vspdemo**

Vspdemo demonstrates the video preview capabilities of the VSP2000 by building a simple DirectShow filter graph to preview the video input. It also demonstrates how to expose the provided Filters Custom Property pages to allow control of the advanced features of the VSP2000.

### Vspcap

VSPCAP demonstrates how to use the VSP2000 to capture a single video frame. This application uses the DirectShow SampleGrabber filter to obtain the frame data and runs in two modes: Overlay Mode and Capture Mode. In overlay mode YUY2 data is captured and displayed using DirectX overlay surfaces. In capture mode RGB data is captured and transferred to the display using Windows GDI functions.



**Vspcap**

# Software Development Kit

## VSP2000 12-Channel Video Surveillance Processor

### Vspgrid

The VspGrid sample application offers similar functionality to VspDemo and displays the video sources in a grid arrangement. It provides a basic GUI for controlling the application.



**Vspgrid**

### Vspmotionview

The VspMotionView application demonstrates the motion detection functionality of the VSP2000. It builds a Directshow graph to preview the captured video and the channel with active motion is automatically displayed on the preview channel. The preview channel can be displayed fullscreen or the last four channels to detect motion can be displayed. Motion detection windows and settings can be configured via the property pages of the VSP2000 custom filters.



**Vspmotionview**

### Vspcapmulti

The Vspcapmulti demonstration application is an expansion of VspCap. It illustrates how to use the VSP2000 to obtain single frame data from multiple VSP2000 cards. The application captures a frame from each of the available inputs in turn. The frames are drawn as thumbnails and a bitmap of the whole is created in the working directory.



**Vspcapmulti**

# Software Development Kit

## VSP2000 12-Channel Video Surveillance Processor

### **The VSP2000 Custom Interface**

The VSP2000 SDK software is provided in the form of a WDM Kernel Streaming driver and supporting KsStreaming plugin filters which expose the AMP custom interfaces through DirectShow. The WDM driver for the VSP2000 provides a standard DirectShow type method of controlling and using the VSP2000 under supported Windows versions. This allows for DirectShow graphs to be built using the VSP2000 as the source for video data.

The VSP2000 software exposes custom DirectShow interfaces for controlling the advanced features of the card.

### **IAMPVSPDecoder**

Control the colour settings of an input: brightness, contrast, hue and saturation. The twelve input channels can be individually controlled.

### **IAMPVSPQuad**

The IAMPVSPQuad custom interface is used to control the display and recording of the VSP2000 input channels. One to twelve inputs can be simultaneously displayed on the preview channel in a variety of arrangements and individual channels can be highlighted with a colored frame.

Channels can be individually paused and zoomed at varying scales.

The IAMPVSPQuad interface also selects which inputs are directed to the record output.

### **IAMPVSPMotion**

The IAMPVSPMotion custom interface controls the motion detection functionality of the VSP2000. It allows the application to specify for each channel the regions within which motion detection will be carried out and the motion detection and blind camera thresholds. Motion detection can be enabled and disabled on individual channels.

The interface indicates when motion is detected on a channel and provides the centre of detected motion. Regions of detected motion can be highlighted on the display by means of a colored tint overlay.

### **IAMPVSPMotionCB**

The IAMPVSPMotionCB interface allows registration of a callback function for receiving notification of motion detection events. The callback function is called whenever motion is detected on one or more enabled channels.





### Creating Applications

In addition to providing sample applications, the VSP2000 SDK contains clear instructions on how to implement a basic video preview application using C++. The header files and library files necessary for utilising the VSP2000 custom interface are provided.

The SDK describes the steps required to create an application and provides example code. It describes how to initialise DirectShow and obtain the pointers to the interfaces and how to build the filter graph which forms the core of the application. Finally, the SDK illustrates how to obtain the pointers to the custom interface and use it to control some of the more advanced features of the VSP2000.