DVS-510

4 Channels 1U Rackmount Digital Video Platform

User Manual

Copyright and Disclaims

Acknowledgements

Intel and Pentium are trademarks of Intel Corporation.
Microsoft Windows and MS-DOS are registered trademarks of Microsoft Corp.
All other product names or trademarks are properties of their respective owners.

May 2007 1st Edition
Printed in Taiwan

Product Warranty (2 year)

Advantech warrants to you, the original purchaser, that each of its products will be free from defects in materials and workmanship for two year from the date of purchase.

This warranty does not apply to any products which have been repaired or altered by persons other than repair personnel authorized by Advantech, or which have been subject to misuse, abuse, accident or improper installation. Advantech assumes no liability under the terms of this warranty as a consequence of such events.

Because of Advantech's high quality-control standards and rigorous testing, most of our customers never need to use our repair service. If an Advantech product is defective, it will be repaired or replaced at no charge during the warranty period. For out-of-warranty repairs, you will be billed according to the cost of replacement materials, service time and freight. Please consult your dealer for more details.

If you think you have a defective product, follow these steps:
1. Collect all the information about the problem encountered. (For example, CPU speed, Advantech products used, other hardware and software used, etc.) Note anything abnormal and list any onscreen messages you get when the problem occurs.

2. Call your dealer and describe the problem. Please have your manual, product, and any helpful information readily available.

3. If your product is diagnosed as defective, obtain an RMA (return merchandize authorization) number from your dealer. This allows us to process your return more quickly.

4. Carefully pack the defective product, a fully-completed Repair and Replacement Order Card and a photocopy proof of purchase date (such as your sales receipt) in a shippable container. A product returned without proof of the purchase date is not eligible for warranty service.

5. Write the RMA number visibly on the outside of the package and
ship it prepaid to your dealer.

Declaration of Conformity

CE
This product has passed the CE test for environmental specifications when shielded cables are used for external wiring. We recommend the use of shielded cables. This kind of cable is available from Advantech. Please contact your local supplier for ordering information.

This product has passed the CE test for environmental specifications. Test conditions for passing included the equipment being operated within an industrial enclosure. In order to protect the product from being damaged by ESD (Electrostatic Discharge) and EMI leakage, we strongly recommend the use of CE-compliant industrial enclosure products.

FCC Class A
Note: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

Technical Support and Assistance

Step 1. Visit the Advantech web site at www.advantech.com/support
where you can find the latest information about the product. Contact your distributor, sales representative, or Advantech's customer service center for technical support if you need additional assistance.

Please have the following information ready before you call:
- Product name and serial number
- Description of your peripheral attachments
- Description of your software (operating system, version, application software, etc.)
- A complete description of the problem
- The exact wording of any error messages

DVS-510 Series Model

Part Number	Video Channel	CPU	HDD
DVS-510-35IKE	4 CH	Support Intel Core Duo Processors	3.5” HDD
* DVS-510-35IKE can support both IDE and SATA HDD.			

Table 0.1 DVS-510 Model List

Packing List

Before installing your board, make sure that the following materials have been received:

Part Number	Description	Quantity
1700001618	VGA Cable 15P to BNCx8 15cm	1
19310305C0	S/S D=5.8 H=1.6 + M3*5L ST BZn NK	6
1960007363	Fix Ear R/L for DVS-510-1U A1	2
1700003413	WIRE 4P/15P BIG4P/SATA POWER 250MM	1
1700002155	CABLE SATA 180D W/LOCK & 90D 30cm	2
1700060202	CABLE 6P-6P-6P 20cm PS/2 KB & MOUSE	1
2066554E00	CD ROM for DVMB-554E V1.00	1
2190000902	CARDBOARD-WARRANTY REV. A2	1

If any of these items are missing or damaged, contact your distributor or sales representative immediately.

Safety Instructions

1. Please read these safety instructions carefully.

2. Please keep this User Manual for later reference.

3. Please disconnect this equipment from power outlet before cleaning. Don‘t use liquid or sprayed detergent for cleaning. Use moisture sheet or clothe for cleaning.

4. For pluggable equipment, the socket-outlet shall near the equipment and shall be easily accessible.

5. Please keep this equipment from humidity.

6. Lay this equipment on a reliable surface when install. A drop or fall could cause injury.

7. Do not leave this equipment in an uncontrolled environment; storage temperatures above 50ºC may damage the equipment.

8. The openings on the enclosure are for air convection hence protecting the equipment from overheating. DO NOT COVER THE OPENINGS.

9. Make sure the voltage of the power source when connecting the equipment to the power outlet.

10. Place the power cord such a way that people cannot step on it. Do not place anything over the power cord. The power cord must be rated for the product and for the voltage and current marked on the product’s electrical ratings label. The voltage and current rating of the cord should be greater than the voltage and current rating marked on the product.

11. All cautions and warnings on the equipment should be noted.

12. If the equipment is not used for long time, disconnect the equipment from mains to avoid being damaged by transient over-voltage.

13. Never pour any liquid into ventilation openings; this could cause fire or electrical shock.

14. Never open the equipment. For safety reasons, only qualified service personnel should open the equipment.

15. If one of the following situations arise, get the equipment checked by service personnel:
   a. The Power cord or plug is damaged.
   b. Liquid has penetrated the equipment.
   c. The equipment has been exposed to moisture.
   d. The equipment has not worked well or you can not get it work according to user‘s manual.
   e. The equipment has been dropped and damaged.
   f. The equipment has obvious signs of breakage

16. This device complies with Part 15 of the FCC rules. Operation is subject to the following two conditions:
   a. this device may not cause harmful interference, and
   b. this device must accept any interference received, including interference that may cause undesired operation.

CAUTION!
THIS COMPUTER IS PROVIDED WITH A BATTERY-POWERED REAL-TIME CLOCK CIRCUIT. THERE IS A DANGER OF EXPLOSION IF BATTERY IS INCORRECTLY REPLACED. REPLACE ONLY WITH SAME OR EQUIVLENT TYPE RECOMMENDED BY THE MANUFACTURE. DISCARD USED BATTERIES ACCORDING TO THE MANUFACTURER’S INSTRUCTIONS.

1

General Information

Chapter 1 Hardware Configuration

1.1 Introduction

The DVS-510 is designed with the Intel® 945GM and the ICH7M-DH for industrial applications that require both high-performance computing and enhanced power management capabilities. The motherboard supports Intel Core™ 2 Duo/Core™ Duo/Core™ Solo Processor up to 2.16GHz with 533/667 MHz front side bus and Dual Channel DDRII 533/667 MHz memory up to 4 GB.

The DVS-510 offers high-performance cost-saving integrated graphics, built on the Intel® 945GM chipset and features the unique Intel® Extreme Graphics architecture that maximizes VGA performance and shares system memory up to 224MB. Better still, the DVS-510 also provides incredible visual quality, versatile display options, 8-bit Dual Channel LVDS and a TV-out(NTSC/PAL) interface.

In addition to powerful computing capabilities, the DVS-510 comes with advanced I/O enhancements. The DVS-510 possess multiple high performance onboard I/O capabilities which include one PCI-Express x4 slot; one Mini-PCI socket; eight high-speed USB 2.0 ports; two Serial ATA ports supporting up to two devices with software Serial ATA RAID 0,1; AC-97 audio; two RS-232 ports; one parallel ATA port. These powerful I/O capabilities ensure even more reliable data storage capabilities and suitable for work with high-speed I/O peripherals.

With all these exceptional features and outstanding performance, DVMB-554 is simply the best, most advanced yet power saving platform for today and tomorrow’s up-and-coming applications.

1.2 Features

- **PCI&PCI Express architecture:** Designed with the Intel 945GM and ICH7M-DH PCI-Express chipset, the DVS-510 has dual/single Gigabit LAN via PCI-E x1 bus, 1 Mini-PCI socket and 1 PCI-E x 4 slot.
- **High Performance I/O Capability:** Mini-PCI and Dual Gigabit LAN via PCI- E x1 bus, 2 SATA connectors and 8 USB 2.0 ports.
- **Standard Mini-ITX form factor with industrial features:** DVS-510 provides industrial features like long product life, reliable operation under wide temperature range, watchdog timer, CMOS backup functions, etc.
- **BIOS CMOS backup and restore:** When BIOS CMOS setup has been completed, data in the CMOS RAM is automatically backed up to the Flash ROM. This is particularly useful in harsh environments which may cause setup data loss such as battery failure. Upon such an error occurring, BIOS will check the data, and automatically restore the original data for booting.
- **Automatically power on after power failure:** It is often required to have an unattended system come back to operation when power resumes after a power failure. Advantech's industrial motherboard allows users to set the system to power on automatically without pushing the power on button.

1.3 Specifications

1.3.1 System

- **CPU:** Socket 479 Intel Core™ 2 Duo/Core™ Duo/Core™ Solo up to 2.16GHz 533/667 MHz FSB.
- **L2 Cache:** CPU has built-in 2MB or 4MB CPU full-speed L2 cache.
- **BIOS:** Award Flash BIOS (4Mb Flash Memory)
- **System Chipset:** Intel 945GM with ICH7M-DH
- **SATA/EIDE hard disk drive interface:** Two on-board SATA connectors with data transmission rate up to 150 MB/s, and supports up to two devices with software Serial ATA RAID 0,1. One on-board IDE connector supporting up to two enhanced IDE devices. Supports PIO mode 4 (16.67MB/s data transfer rate) and ATA 33/66/100 (33/66/100MB/s data transfer rate.) BIOS enabled/disabled.

1.3.2 Memory

- **RAM:** Up to 4 GB in four 240-pin DIMM sockets. Supports dual-chan- nel DDRII 400/533/667 SDRAM.

1.3.3 Input/Output

- **PCI Express slots:** 1 PCI-E x 3 with PCI-E x 4 expansion slot (this expansion slot works with Advantech Riser card) and 1 PCI-E x 1 gold finger.
- **PCI Bus:** 1 Mini-PCI socket, 32-bit, 33 MHz PCI 2.2 compliant
- **Serial ports:** Two serial ports, one DB-9 (RS 232)connector and one on-board pin header (RS 232/422/485). Ports can be individually con- figured to COM1, COM2, or disabled
- **Keyboard and PS/2 mouse connector:** One 6-pin Mini-DIN connec- tors are located on the mounting bracket and work with special Y cable for easy connection to a PS/2 keyboard and mouse.
- **USB port:** Supports up to eight USB 2.0 ports with transmission rate up to 480Mbps.

1.3.4 Graphics

- **Controller:** Intel Graphics Media Accelerator 950
- **Display memory:** Dynamically shared system memory up to 224 MB.
- **VGA:** Up to 2048x1536 resolution@75Hz
- **LVDS interface:** Support up to UXGA(1600X1200)
- **TV-Out:** NTSC/PAL.

1.3.5 Ethernet LAN

- Supporting single/dual 10/100/1000Base-T Ethernet port(s) via PCI Express x1 bus which provides 500 MB/s data transmission rate.
- **Controller:**

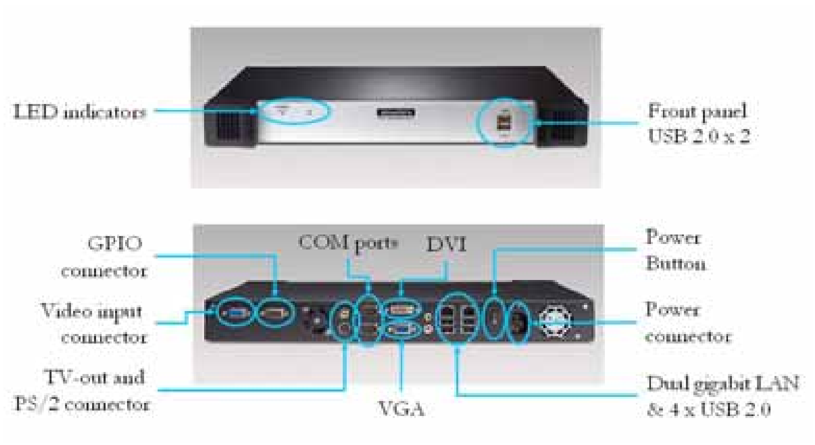
  Marvell 88E8053 PCI-E Gigabit LAN; featuring AI NET2

1.3.6 Industrial features

- **Watchdog timer:** Can generate a system reset or IRQ11. The watch-dog timer is programmable, with each unit equal to one second or one minute (255 levels)

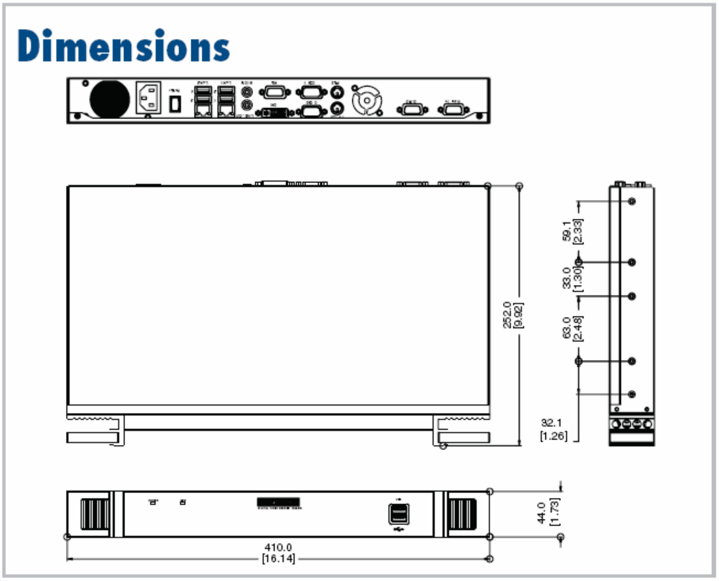
1.3.7 Mechanical and environmental specifications

- **Operating temperature:** -15° C ~ 50° C (Depending on CPU)
- **Power supply:** AC input 100~240VAC, 4A, 60-50Hz, 180W
- **Dimensions:** 410 x 44 x 252 mm



1.3.8 Rockmount structure

• DVS-510-35IKE is 1U based rockmount platform. There is a mounting kit on accessory box.



1.4 Jumpers and Connectors

Connectors on the DVS-510 motherboard link it to external devices such as hard disk drives and a keyboard. In addition, the board has a num- ber of jumpers used to configure your system for your application.

The tables below list the function of each of the board jumpers and con- nectors. Later sections in this chapter give instructions on setting jump- ers. Chapter 2 gives instructions for connecting external devices to your motherboard.

Label	Function
J2	COM2 RS232/422/485 selectors
J3	CMOS Clear
J4	Watchdog timer output selection
J5	AT/ATX mode selector
J6	LVDS Power 3.3V/5V selector

Label	Function
CM1	Serial port module: COM1/COM2
CN2	External PS/2 and Composite TV-Out
CN3	Line Out/MIC In connector
CN4	LAN2; USB ports 3, 4
CN5	LAN1; USB ports 1, 2
CN6	Rear Panel audio connector
CN7	AUX-IN connector
CN8	IrDA connector
CN9	USB ports 5, 6
CN10	USB ports 7, 8
CN11	Primary IDE connector
CN12	Hardware Monitor connector
CN13	Power LED

Label	Function
CNX1	Power/Reset/HDD LED/Alarm specker connector
GPIO1	GPIO connector
VDO1	DVA-210 connector
FN1	FAN connector
FN2	FAN connector
FN3	FAN connector
SMBUS1	SMBus Extend connector
SA1	Serial ATA1
SA2	Serial ATA2
VCN1	LVDS connector
BKL1	LCD Inverter Power/Back light connector
V1	VGA and DVI connector module
ATX1	ATX 12 V auxiliary power connector ATX2
	20-pin ATX power connector

1.5 Jumper and Connector Locations

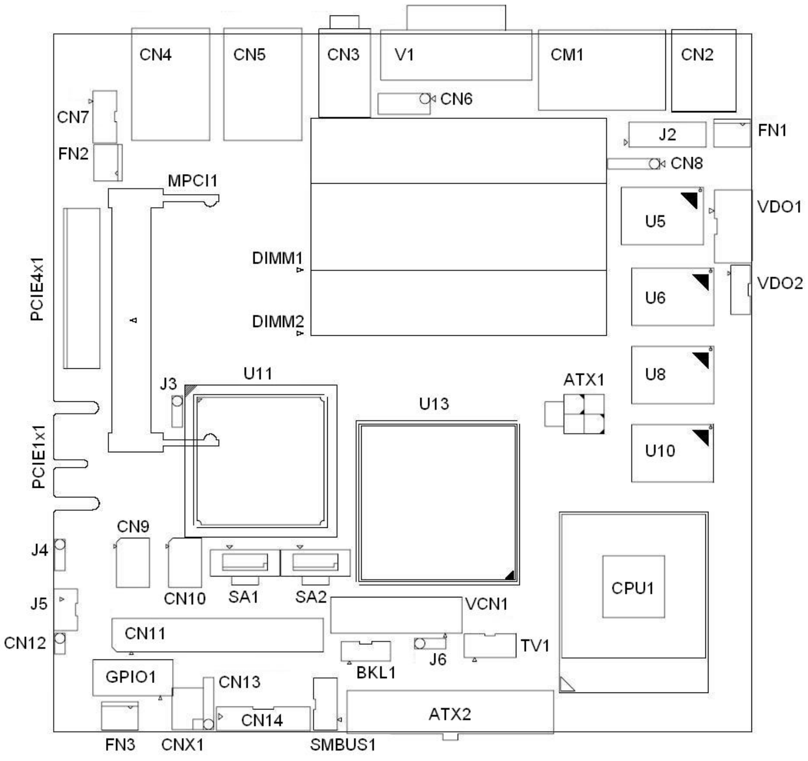


Figure 1.1: Jumper and Connector locations

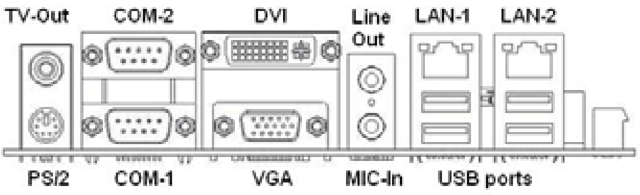


Figure 1.2: Rear Panel Placement

1.6 DVS-510 Block Diagram

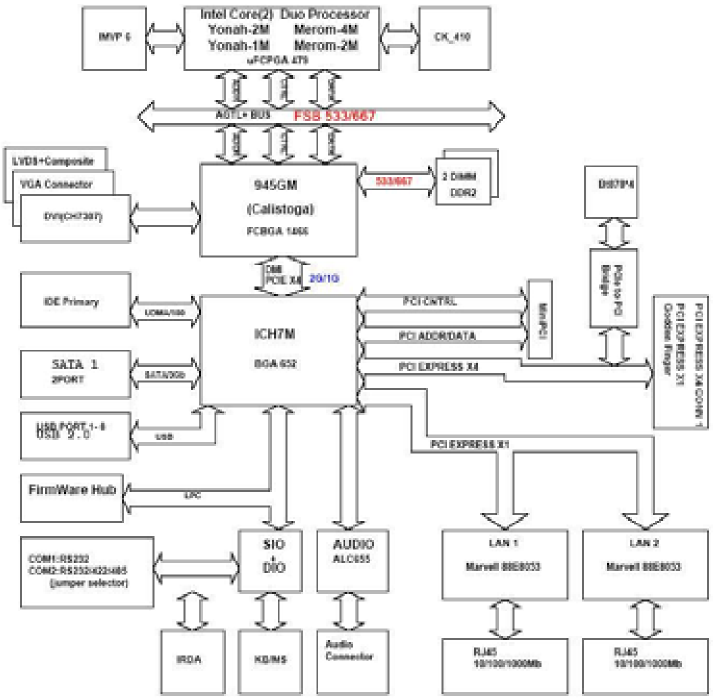


Figure 1.3: DVS-510 Block Diagram

1.7 Safety Precautions

Warning! *Always completely disconnect the power cord from your chassis whenever you work with the hardware. Do not make connections while the power is on. Sensitive electronic components can be damaged by sudden power surges. Only experienced electronics personnel should open the PC chassis.*

Caution! *Always ground yourself to remove any static charge before touching the motherboard. Mod- ern electronic devices are very sensitive to static electric charges. As a safety precaution, use a grounding wrist strap at all times. Place all electronic components on a static-dissipative surface or in a static-shielded bag when they are not in the chassis.*

Caution! *The computer is provided with a battery-pow- ered Real-time Clock circuit. There is a danger of explosion if battery is incorrectly replaced. Replace only with same or equivalent type rec- ommended by the manufacturer. Discard used batteries according to manufacturer's instruc- tions.*

Caution! *There is a danger of a new battery exploding if it is incorrectly installed. Do not attempt to recharge, force open, or heat the battery. Replace the battery only with the same or equivalent type recommended by the manufac- turer. Discard used batteries according to the manufacturer’s instructions.*

1.8 Jumper Settings

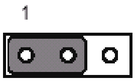
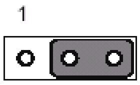
This section provides instructions on how to configure your motherboard by setting the jumpers. It also includes the motherboards's default settings and your options for each jumper.

1.8.1 How to set jumpers

You can configure your motherboard to match the needs of your application by setting the jumpers. A jumper is a metal bridge that closes an electrical circuit. It consists of two metal pins and a small metal clip (often protected by a plastic cover) that slides over the pins to connect them. To "close" (or turn ON) a jumper, you connect the pins with the clip. To "open" (or turn OFF) a jumper, you remove the clip. Sometimes a jumper consists of a set of three pins, labeled 1, 2, and 3. In this case you connect either pins 1 and 2, or 2 and 3. A pair of needle-nose pliers may be useful when setting jumpers.

1.8.2 CMOS clear (J3)

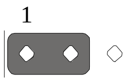
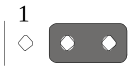
The DVS-510 motherboard contains a jumper that can erase CMOS data and reset the system BIOS information. Normally this jumper should be set with pins 1-2 closed. If you want to reset the CMOS data, set J1 to 2-3 closed for just a few seconds, and then move the jumper back to 1-2 closed. This procedure will reset the CMOS to its default setting.

Function	Jumper Setting	
* Keep CMOS data		1-2 closed
Clear CMOS data		2-3 closed

* default setting

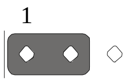
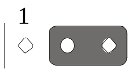
1.8.3 Watchdog timer output (J4)

The DVS-510 contains a watchdog timer that will reset the CPU or send a signal to PIRQE in the event the CPU stops processing. This feature means the DVS-510 will recover from a software failure or an EMI problem. The J4 jumper settings control the outcome of what the computer will do in the event the watchdog timer is tripped.

Function	Jumper Setting
IRQ11	 1-2 closed
* Reset	 2-3 closed

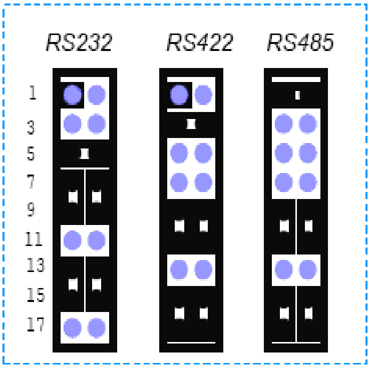
*default setting

Note: *The interrupt output of the watchdog timer is a low level signal. It will be held low until the watchdog timer is reset.*

Function	Jumper Setting
AT Mode	 1-2 closed
ATX Mode	 2-3 closed

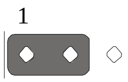
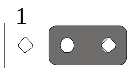
1.8.4 COM2 RS 232/422/485 mode selector (J2)

Users can use J2 to select among RS 232/422/485 modes for COM2. The default setting is RS 232.



Function	Jumper Setting
RS232	(5-6) + (7-9) + (8-10) + (13-15) + (14-16) closed RS422
	(3-4) + (9-11) + (10-12) + (15-17) + (16-18) closed
RS485	(1-2) + (9-11) + (10-12) + (15-17) + (16-18) closed

1.8.5 LVDS Power 3.3V/5V selector (J6)

Function	Jumper Setting	
5V		1-2 closed
*3.3V		2-3 closed

*default setting

1.9 System Memory

The DVS-510 has two sockets for 240-pin memory modules (SO-DIMMs). All these sockets use 1.8 V unbuffered double data rate synchronous DRAMs (DDR SDRAM). They are available in capacities of 256, 512 and 1024 MB. The sockets can be filled in any combination with DIMMs of any size, giving a total memory size between 256 MB and 2GB.

1.9.1 CPU FSB and memory speed

The DVS-510 can accept DDR2 SDRAM memory chips without parity. Also note: The DVS-510 accepts DDR2 400/533/667MHz SDRAM, and DDR2 SDRAM. The DVS-510 does NOT support ECC (error checking and correction).

1.10 Memory Installation Procedures

To install DIMMs, first make sure the two handles of the DIMM socket are in the “open” position. i.e. The handles lean outward. Slowly slide the DIMM module along the plastic guides on both ends of the socket. Then press the DIMM module right down into the socket, until you hear a click. This is when the two handles have automatically locked the mem- ory module into the correct position of the DIMM socket. To remove the memory module, just push both handles outward, and the memory mod- ule will be ejected by the mechanism in the socket.

1.11 Cache Memory

The CPU that DVS-510 supports built-in 2MB or 4MB cache memory.

1.12 Processor Installation

Warning: *Without a fan or heat sink, the CPU will over- heat and cause damage to both the CPU and the single board computer. To install a CPU, first turn off your system. Locate the processor socket 479.*

The DVS-510 is designed for Intel Core™ 2 Duo/Core™ Duo/Core™ Solo (socket 479) up to 2.16 GHz. Follow these steps to install the processor:

1. Turn the screw to loosen the processor socket.
2. Align the triangular marking on the processor with the small arrow on the corner of the socket.
3. Turn the screw to its original position.
4. Install the heat sink on the CPU. The concave of heat sink should face to location of capacitors to avoid contacting with capacitors.

2

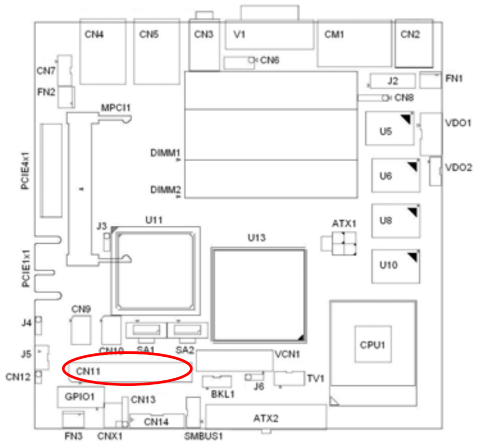
Connecting Peripherals

Chapter 2 Connecting Peripherals

2.1 Introduction

You can access most of the connectors from the top of the board as it is being installed in the chassis. If you have a number of cards installed or have a packed chassis, you may need to partially remove the card to make all the connections.

2.2 Primary IDE Connector (CN11)



You can attach up to two IDE (Integrated Drive Electronics) drives to the DVS-510’s built-in controller. The primary (CN11) connector can each accommodate two drives.

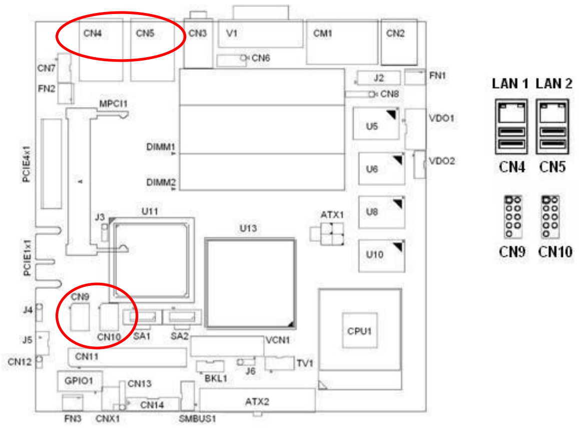
Wire number 1 on the cable is red or blue and the other wires are gray. Connect one end to connector CN11 on the motherboard. Make sure that the red/blue wire corresponds to pin 1 on the connector (in the upper right hand corner). See Chapter 1 for help finding the connector.

Unlike floppy drives, IDE hard drives can connect in either position on the cable. If you install two drives to a single connector, you will need to set one as the master and the other as the slave. You do this by setting the jumpers on the drives. If you use just one drive on the connector, you should set the drive as the master. See the documentation that came with your drive for more information.

Connect the first hard drive to the other end of the cable. Wire 1 on the cable should also connect to pin 1 on the hard drive connector, which is labeled on the drive circuit board. Check the documentation that came with the drive for more information.

2.3 USB Ports and LAN Ports (CN4, CN5, CN9, CN10)

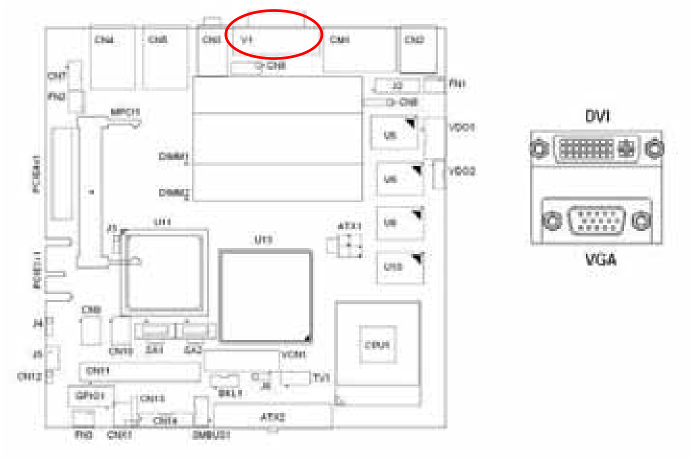
The DVS-510 provides up to eight ports of USB (Universal Serial Bus) interface which gives complete Plug & Play and hot swapping for up to 127 external devices. The USB interface complies with USB Specification Rev. 2.0 supporting transmission rate up to 480 Mbps and is fuse protected. The USB interface can be disabled in the system BIOS setup.



The DVS-510 is equipped with one or two high-performance 1000 Mbps Ethernet LANs. They are supported by all major network operating systems. The RJ-45 jacks on the rear plate provide convenient or 1000 Base-T operation.

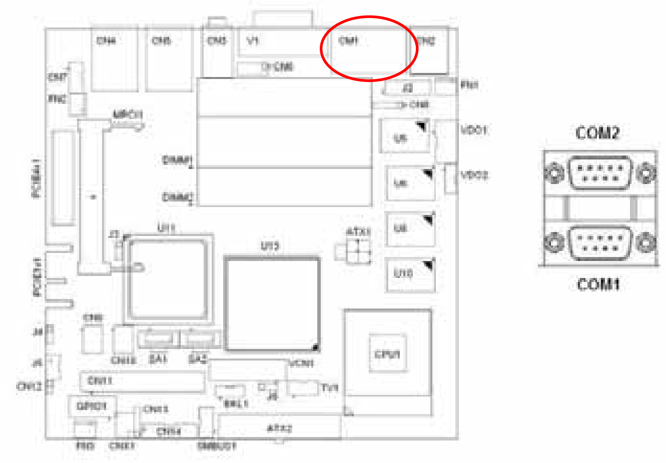
Lan mode	**Lan Indicator**
1Gbps Link on	LED1 Green on
100Mbps Link on	LED1 Orange on
Active	LED2 Green flash

2.4 VGA and DVI connector module (V1)



The DVS-510 includes DVI and VGA interface that can drive dual displays.

2.5 Serial port module: COM1/COM2 (CM1)

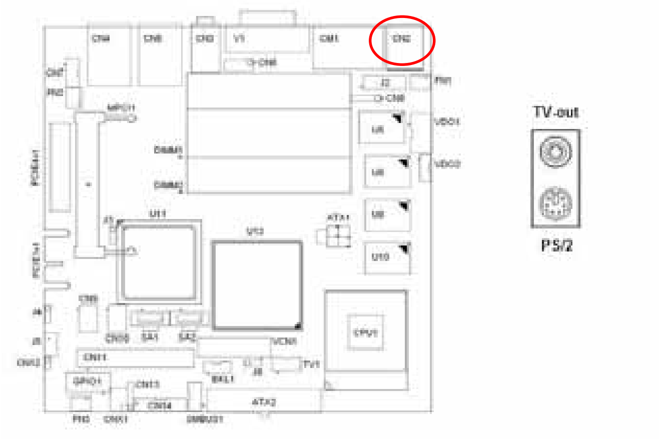


The DVS-510 offers two serial ports. The user can use J2 to select among RS 232/422/485 modes for COM2. The default setting is RS 232 for both COM1 and COM2. These ports can connect to serial devices, such as a mouse or printer, or to a communications network.

The IRQ and address ranges for both ports are fixed. However, if you want to disable the port or change these parameters later, you can do this in the system BIOS setup.

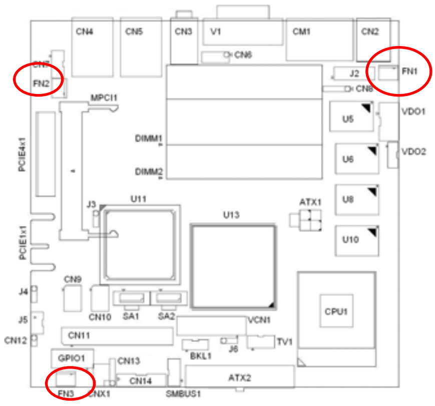
Different devices implement the RS-232/422/485 standards in different ways. If you are having problems with a serial device, be sure to check the pin assignments for the connector.

2.6 External PS/2 and Composite TV-Out (CN2)



The DVS-510 provides a PS/2 keyboard/mouse connector. A 6-pin mini-DIN connector is located on the rear face plate. It comes with an external Y cable to convert from the 6-pin mini-DIN connector to PS/2 keyboard and PS/2 mouse connection.

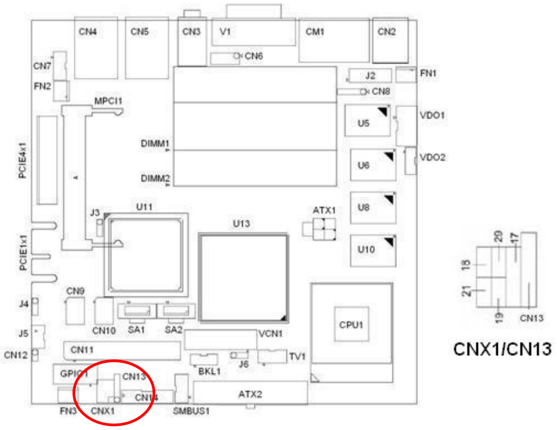
2.7 Fan Connector (FN1/FN2/FN3)



If fan is used, this connector supports cooling fans of 500 mA (6W) or less.

2.8 Front Panel Connectors (CNX1/CN13)

There are several external switches to monitor and control the DVMB-554.



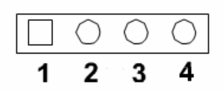
2.8.1 Power LED and Keyboard Lock (CN13)

CN13 is a 5-pin connector for the power LED. Refer to Appendix B for detailed information on the pin assignments. If a PS/2 or ATX power sup- ply is used, the system's power LED status will be as indicated below:

Power mode	LED (PS/2 power)	LED (ATX power)
System On	On	On
System Suspend	Fast flashes	Fast flashes
System Off	Off	Slow flashes

2.8.2 External Speaker (CNX1-17)

It is a 4-pin connector for an external speaker. If there is no external speaker, the DVS-510 provides an onboard buzzer as an alternative. To enable the buzzer, set pins 3-4 as closed.



2.8.3 Reset Connector (CNX1-18)

Many computer cases offer the convenience of a reset button. Connect the wire from the reset button

2.8.4 HDD LED Connector (CNX1-19)

You can connect an LED to this connector to indicate when the HDD is active.

2.8.5 ATX Soft Power Switch (CNX1-21)

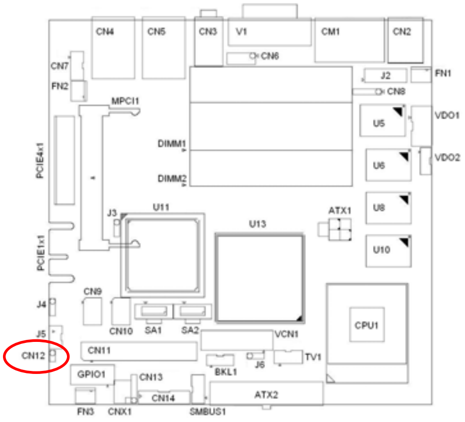
If your computer case is equipped with an ATX power supply, you should connect the power on/off button on your computer case to this connector. This connection enables you to turn your computer on and off.

2.8.6 SM Bus Connector (CNX1-29)

This connector is reserved for Advantech's SNMP-1000 HTTP/SNMP Remote System Manager. The SNMP-1000 allows users to monitor the internal voltages, temperature and fans from a remote computer through an Ethernet network.

CN29 can be connected to CN19 of SNMP-1000. Please be careful about the pin assignments, pin 1 must be connected to pin 1 and pin 2 to pin 2 on both ends of cable.

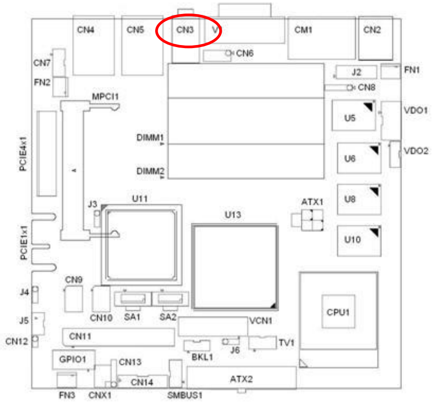
2.9 H/W Monitor Alarm (CN12)



Close: Enable OBS Alarm

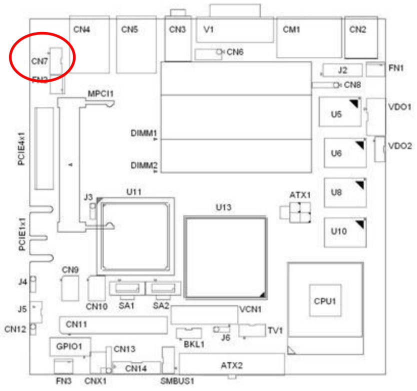
Open: Disable OBS Alarm

2.10 Line Out, Mic In Connector (CN3)



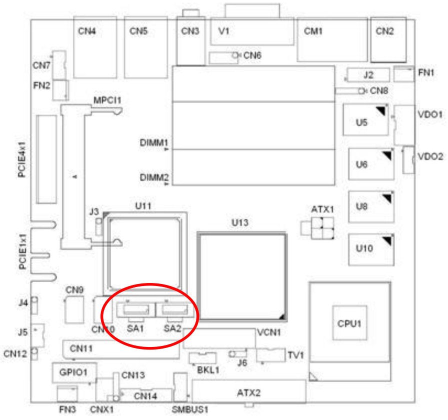
The Line Out is to output the audio signal to external audio device, like speakers or headphones. The Mic In is for the audio signal input via microphones.

2.11 Aux Line-In Connector (CN7)



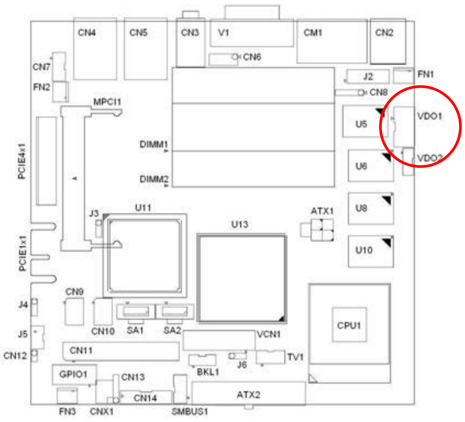
The connector is for audio devices with a Line-in connector.

2.12 Serial ATA Interface (SA1, SA2)



In addition to the EIDE interface (up to two devices), the DVS-510 features a high performance serial ATA interface (up to 150 MB/s) which eases cabling to hard drives with thin and long cables.

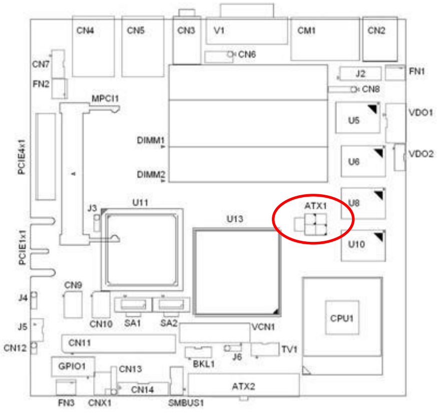
2.13 DVA-210 Connector (VDO1)



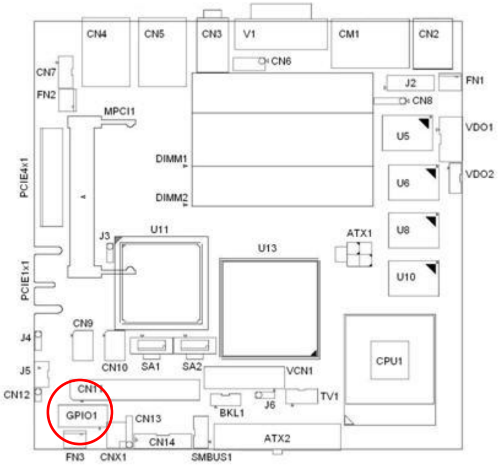
DVS-510 has 4 video capture chips on board. So it can support up to 4 cameras. To be connecting with cameras, DVS-510 needs to work with DVA-210 for video capture. DVA-210 is a 4 channel video module with BNC connectors and you can find it in DVS-510 accessory box.

2.14 Auxiliary 4-pin power connector (ATX1)

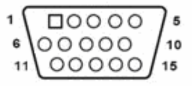
To ensure the enough power is supplied to the motherboard, one auxiliary 4 pin power connector is available on the DVS-510. ATX1 must be used to provide sufficient 12 V power to ensure the stable operation of the system.



2.15 GPIO connector (GPIO1)



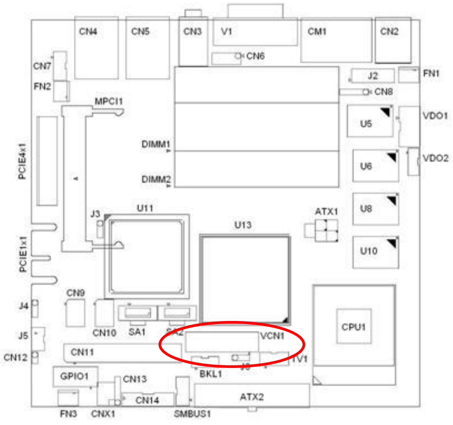
The DVS-510 provides a GPIO interface, GPIO1, which is a 20 pin connector. Advantech provides SDK for this GPIO and have 6 pins for input and 8 pins for output (7 pins output for DVS-510-35IKE). Please refer to pin assignments for GPIO1 connector as below.



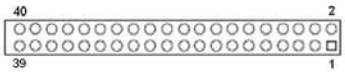
GPIO Pin assignment of DVS-510-35IKE

GPIO connector (GPIO1)							
Pin	**signal**	**pin**	**signal**	**pin**	**signal**	**pin**	**signal**
1	+5V	6	Input	11	Output	16	Input
2	+5V	7	Output	12	Input	17	Output
3	Output	8	Input	13	Output	18	GND
4	Input	9	Output	14	Input	19	GND
5	Output	10	Input	15	Output	20	GND

2.16 LVDS connector (VCN1)

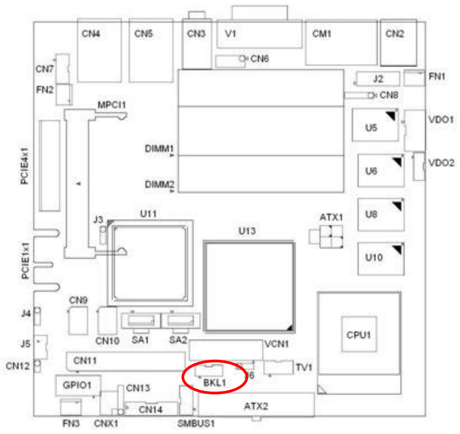


The DVS-510 provides a LVDS interface supports 18 bits LCD panels. Pin assignments for LVDS connector VCN1 are listed as below.



LVDS connector (VCN1)							
Pin	**signal**	**pin**	**signal**	**pin**	**signal**	**pin**	**signal**
1	VDDS AFE	11	GND	21	OD2+	31	DDC_ CLK
2	VDDS AFE	12	GND	22	OD2+	32	DDC_ DAT
3	GND	13	OD1-	23	GND	33	GND
4	GND	14	OD1-	24	GND	34	GND
5	VDDS AFE	15	OD1+	25	OCK-	35	OD3-
6	VDDS AFE	16	OD1+	26	OCK-	36	OD3-
7	OD0-	17	GND	27	OCK+	37	OD3+
8	OD0-	18	GND	28	OCK+	38	OD3+
9	OD0+	19	OD2-	29	GND	39	HPLG
10	OD0+	20	OD2-	30	GND	40	VCON

2.17 LCD Inverter Power connector (BKL1)



VP1 is connected to Inverter which can provide power to LCD Panel. The DVS-510 can provide a LVDS display.

3

BIOS Setup

Chapter 3 Award BIOS Setup

3.1 Introduction

Award’s BIOS ROM has a built-in setup program that allows users to modify the basic system configuration. This type of information is stored in battery backed-up memory (CMOS RAM) so that it retains the setup information when the power is turned off.

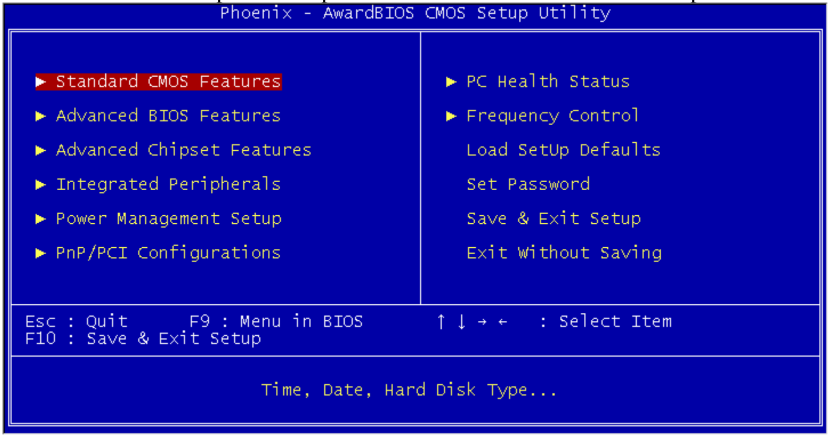
3.1.1 CMOS RAM Auto-backup and Restore

The CMOS RAM is powered by an onboard button cell battery. When you finish BIOS setup, the data in CMOS RAM will be automatically backed up to Flash ROM. If operation in harsh industrial environments causes a soft error, BIOS will recheck the data in CMOS RAM and automatically restore the original data in Flash ROM to CMOS RAM for booting.

Note: *If you intend to change the CMOS setting with- out restoring the previous backup, you have to click on “DEL” within two seconds of the “CMOS checksum error...” display screen message appearing. Then enter the “Setup” screen to modify the data. If the “CMOS checksum error...” message appears again and again, please check to see if you need to replace the battery in your system.*

3.2 Entering Setup

Turn on the computer and press <Del> to enter the BIOS setup.



```
                    Phoenix - AwardBIOS CMOS Setup Utility

 ▶ Standard CMOS Features              ▶ PC Health Status
 ▶ Advanced BIOS Features              ▶ Frequency Control
 ▶ Advanced Chipset Features             Load Setup Defaults
 ▶ Integrated Peripherals                Set Password
 ▶ Power Management Setup                Save & Exit Setup
 ▶ PnP/PCI Configurations                Exit Without Saving

 Esc : Quit       F9 : Menu in BIOS      ↑ ↓ → ←   : Select Item
 F10 : Save & Exit Setup

                       Time, Date, Hard Disk Type...
```

Figure 3.1: Award BIOS Setup initial screen

3.3 Standard CMOS Setup

3.3.1 Date

The date format is <week>, <month>, <day>, <year>.

3.3.2 Time

The time format is <hour> <minute> <second>, based on the 24-hour clock.

3.3.3 IDE channel 0/1 Master/Slave

- **IDE HDD Auto-Detection**: Press “Enter” to select this option for automatic device detection.
- **IDE Device Setup**:
  Auto: Automatically detects IDE devices during POST
  None: Select this when no IDE device is used. The system will skip the auto-detection step to make system start up faster.
  Manual: User can manually input the correct settings.
- **Access Mode**: The options are CHS/LBA/Large/Auto
- **Capacity**: Capacity of currently installed hard disk.
- **Cylinder**: Number of cylinders

- **Head**: Number of heads
- **Precomp**: Write precomp
- **Landing Zone**: Landing zone
- **Sector**: Number of sectors

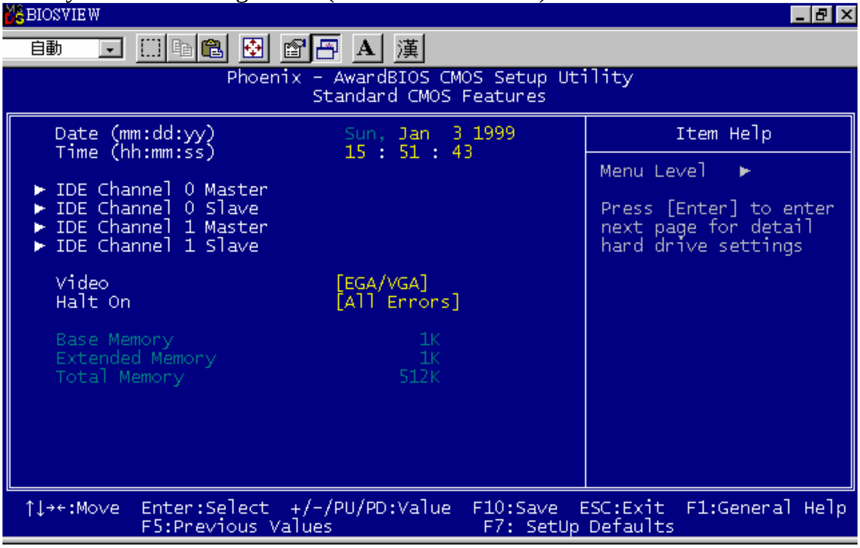
3.3.4 Halt On

This category determines whether system start-up will halt or not when an error is detected during power up.

The options are: No Errors/All Errors/All, But Keyboard/All, But Diskette/All, But Disk/Key

3.3.5 Memory

This category displays base memory, extended memory, and total memory detected during POST (Power On Self Test).

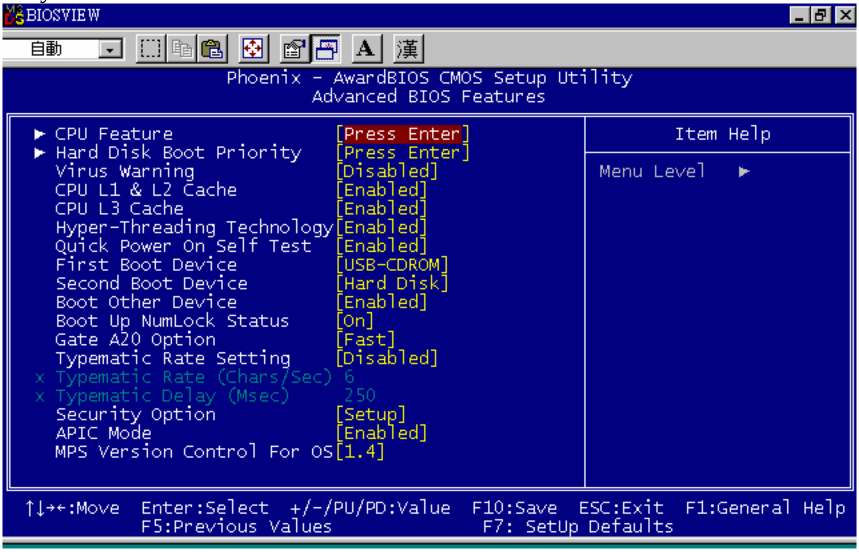


```
                    Phoenix - AwardBIOS CMOS Setup Utility
                           Standard CMOS Features

    Date (mm:dd:yy)              Sun, Jan  3 1999      Item Help
    Time (hh:mm:ss)              15 : 51 : 43
                                                       Menu Level   ▶
  ▶ IDE Channel 0 Master
  ▶ IDE Channel 0 Slave                                Press [Enter] to enter
  ▶ IDE Channel 1 Master                               next page for detail
  ▶ IDE Channel 1 Slave                                hard drive settings

    Video                       [EGA/VGA]
    Halt On                     [All Errors]

    Base Memory                        1K
    Extended Memory                    1K
    Total Memory                     512K

 ↑↓→←:Move  Enter:Select  +/-/PU/PD:Value  F10:Save  ESC:Exit  F1:General Help
            F5:Previous Values                F7: SetUp Defaults
```

Figure 3.2: Standard CMOS Features Screen

3.4 Advanced BIOS Features

The "Advanced BIOS Features" screen appears when choosing the "Advanced BIOS Features" item from the "Initial Setup Screen" menu. It allows the user to configure the DVS-510 according to his particular requirements. Below are some major items that are provided in the Advanced BIOS Features screen. A quick booting function is provided for your convenience. Simply enable the Quick Booting item to save yourself valuable time.



```
              Phoenix - AwardBIOS CMOS Setup Utility
                      Advanced BIOS Features

 ► CPU Feature                   [Press Enter]        Item Help
 ► Hard Disk Boot Priority       [Press Enter]
   Virus Warning                 [Disabled]      Menu Level   ►
   CPU L1 & L2 Cache             [Enabled]
   CPU L3 Cache                  [Enabled]
   Hyper-Threading Technology    [Enabled]
   Quick Power On Self Test      [Enabled]
   First Boot Device             [USB-CDROM]
   Second Boot Device            [Hard Disk]
   Boot Other Device             [Enabled]
   Boot Up NumLock Status        [On]
   Gate A20 Option               [Fast]
   Typematic Rate Setting        [Disabled]
 x Typematic Rate (Chars/Sec)     6
 x Typematic Delay (Msec)         250
   Security Option               [Setup]
   APIC Mode                     [Enabled]
   MPS Version Control For OS    [1.4]

 ↑↓→←:Move   Enter:Select   +/-/PU/PD:Value   F10:Save   ESC:Exit   F1:General Help
             F5:Previous Values                F7: SetUp Defaults
```

Figure 3.3: Advanced BIOS features screen

3.4.1 CPU Features

Delay Prior to thermal

This feature controls the activation of the Thermal Monitor's automatic mode. It allows you to determine when the Pentium 4's Thermal Monitor should be activated in automatic mode after the system boots. For example, with the default value of 16 Minutes, the BIOS activates the Thermal Monitor in automatic mode 16 minutes after the system starts booting up. The choices are 4 Min, 8 Min, 16 Min, and 32 Min.

C1E Function

CPU C1E Function select. Default value is Auto

Execute Disable Bit

When disabled, forces the XD feature flag to always return 0.

3.4.2 Hard Disk Boot Priority

Set hard disk boot device priority.

3.4.3 Virus Warning

Enables or disables the virus warning.

3.4.4 CPU L1, L2 & L3 Cache

Enabling this feature speeds up memory access. The commands are "Enabled" or "Disabled."

3.4.5 Quick Power On Self Test

This allows the system to skip certain tests to speed up the boot-up procedure.

3.4.6 First/Second Boot Device

The BIOS tries to load the OS from the devices in the sequence set here. The options are: "Floppy", "LS120", "CDROM", "Hard Disk", "ZIP100", "USB-FDD", "USB-ZIP", "USBCDROM", "LAN1", "LAN2" and "Disabled".

3.4.7 Boot Other Device

Use this to boot another device. The options are "Enabled" and "Disabled".

3.4.8 Boot Up NumLock Status
Sets the boot up status Num Lock. The options are “On” and “Off”.

3.4.9 Gate A20 Option
“Normal”: A pin in the keyboard controller controls GateA20. Fast (Default) lets chipset control GateA20.

3.4.10 Typematic Rate Setting
The typematic rate is the rate key strokes repeat as determined by the keyboard controller. The commands are “Enabled” or “Disabled”. Enabling allows the typematic rate and delay to be selected.

3.4.11` Typematic Rate (Chars/Sec)
The BIOS accepts the following input values (characters/second) for typematic rate: 6, 8, 10, 12, 15, 20, 24, and 30.

3.4.12 Typematic Delay (msec)
Typematic delay is the time interval between the appearances of two consecutive characters, when the key is continuously depressed. The input values for this category are: 250, 500, 750, and 1000 (ms).

3.4.13 Security Option
This category determines whether the password is required when the system boots up or only when entering setup. The options are:

- **System**: The system will not boot, and access to Setup will be also denied unless the correct password is entered at the prompt.
- **Setup**: The system will boot, but access to Setup will be denied unless the correct password is entered at the prompt.

Note: *To disable security, select PASSWORD SET- TING in the main menu. Then, you will be asked to enter a password. Simply press <Enter> to disable security. When security is disabled, the system will boot and you can enter Setup freely.*

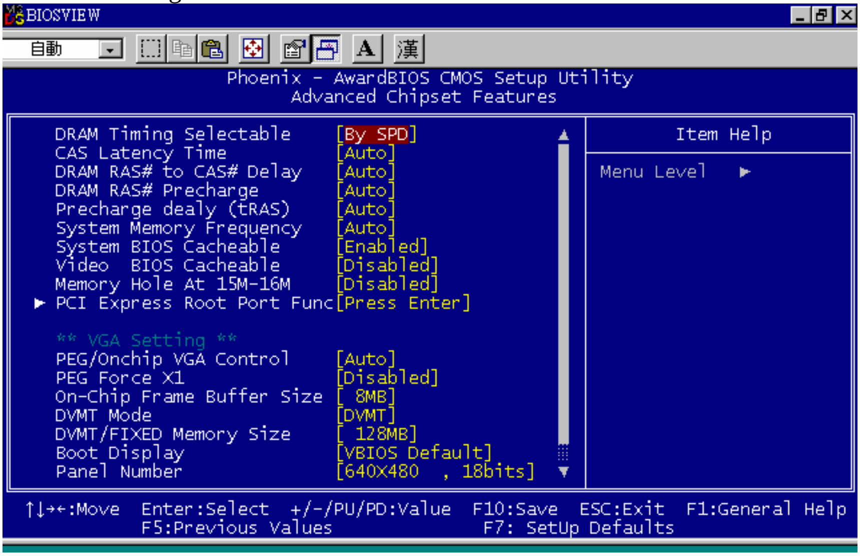
3.4.14 APIC Mode
This setting allows you to enable the APIC mode. The choices are “Disabled” or “Enabled.”

3.4.15 MPS Version Control For OS

This specifies the version of the Multiprocessor Specification (MPS) to be used. Version 1.4 has extended configuration tables to improve support for multiple PCI bus configurations and provide future expandability - use this for NT, and possibly Linux. It is also required for a secondary PCI bus to work without the need for a bridge. Leave it as 1.1 for older server Operating Systems.

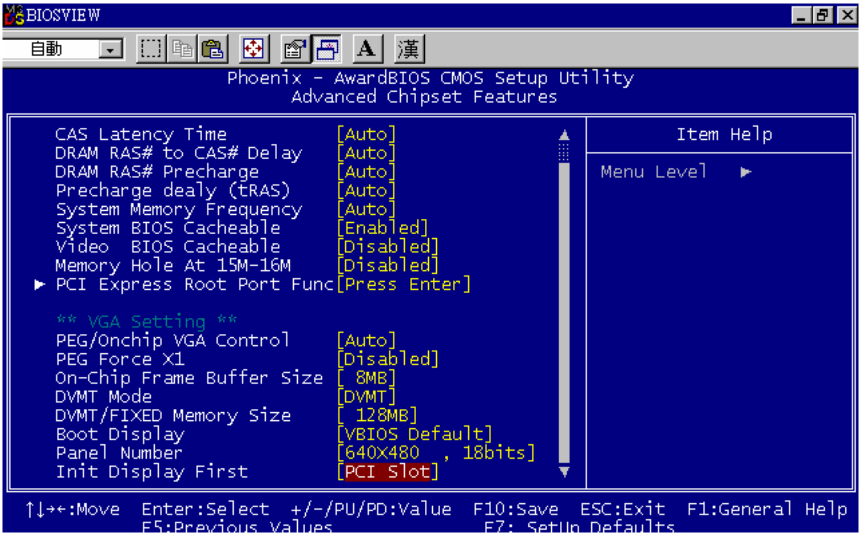
3.5 Advanced Chipset Features

By choosing the “Advanced Chipset Features” option from the “Initial Setup Screen” menu, the screen below will be displayed. This sample screen contains the manufacturer’s default values for the DVS-510, as shown in Figure 3-5:



```
              Phoenix - AwardBIOS CMOS Setup Utility
                     Advanced Chipset Features

    DRAM Timing Selectable        [By SPD]              Item Help
    CAS Latency Time              [Auto]
    DRAM RAS# to CAS# Delay       [Auto]           Menu Level   ►
    DRAM RAS# Precharge           [Auto]
    Precharge dealy (tRAS)        [Auto]
    System Memory Frequency       [Auto]
    System BIOS Cacheable         [Enabled]
    Video  BIOS Cacheable         [Disabled]
    Memory Hole At 15M-16M        [Disabled]
  ► PCI Express Root Port Func[Press Enter]

    ** VGA Setting **
    PEG/Onchip VGA Control        [Auto]
    PEG Force X1                  [Disabled]
    On-Chip Frame Buffer Size     [ 8MB]
    DVMT Mode                     [DVMT]
    DVMT/FIXED Memory Size        [ 128MB]
    Boot Display                  [VBIOS Default]
    Panel Number                  [640X480   , 18bits]

 ↑↓→←:Move  Enter:Select  +/-/PU/PD:Value  F10:Save  ESC:Exit  F1:General Help
            F5:Previous Values                F7: SetUp Defaults
```

Figure 3.5: Advanced chipset features screen (a)



```
Phoenix - AwardBIOS CMOS Setup Utility
       Advanced Chipset Features

     CAS Latency Time              [Auto]                 Item Help
     DRAM RAS# to CAS# Delay       [Auto]
     DRAM RAS# Precharge           [Auto]            Menu Level   ►
     Precharge dealy (tRAS)        [Auto]
     System Memory Frequency       [Auto]
     System BIOS Cacheable         [Enabled]
     Video  BIOS Cacheable         [Disabled]
     Memory Hole At 15M-16M        [Disabled]
   ► PCI Express Root Port Func[Press Enter]

     ** VGA Setting **
     PEG/Onchip VGA Control        [Auto]
     PEG Force X1                  [Disabled]
     On-Chip Frame Buffer Size     [ 8MB]
     DVMT Mode                     [DVMT]
     DVMT/FIXED Memory Size        [ 128MB]
     Boot Display                  [VBIOS Default]
     Panel Number                  [640X480  , 18bits]
     Init Display First            [PCI Slot]

 ↑↓→←:Move  Enter:Select  +/-/PU/PD:Value  F10:Save  ESC:Exit  F1:General Help
            F5:Previous Values                  F7: Setup Defaults
```

Figure 3.5: Advanced chipset features screen (b)

Note: *DRAM default timings have been carefully cho- sen and should ONLY be changed if data is being lost. Please first contact technical sup- port.*

3.5.1 DRAM Timing Selectable

This item allows you to control the DRAM speed. The selections are "Manual" or "By SPD".

3.5.2 CAS Latency Time

When DRAM Timing Selectable is set to [Manual], this field is adjustable. This controls the CAS latency, which determines the time interval between SDRAM starting a read command and receiving it. The options are [3T], [4T], [5T], and [Auto].

3.5.3 DRAM RAS# to CAS# Delay

When DRAM Timing selectable is set to [Manual], this field is adjustable. When DRAM is refreshed, the rows and columns are addressed separately. This setup item allows user to determine the timing of the transition from RAS (row address strobe) to CAS (column address strobe). The less the clock cycles are, the faster the DRAM speed is. Set- ting options are [2T] to [5T], and [Auto].

3.5.4 DRAM RAS# Precharge

When the DRAM Timing Selectable is set to [Manual], this field is adjustable. This setting controls the number of cycles for Row Address Strobe (RAS) to be allowed to precharge. If no sufficient time is allowed for the RAS to accumulate its charge before DRAM refresh, refreshing may be incomplete and DRAM may fail to retain data. This item applies only when synchronous DRAM is installed in the system. Setting options are [2T] to [5T], and [Auto].

3.5.5 Precharge Delay (t RAS)

This item allows you to select the value in this field, depending on whether the board has paged DRAMs or EDO (extended data output) DRAMs. The choices are: “4” to “15” and “Auto”.

3.5.6 System Memory Frequency

To adjust the frequency of memory. The choices are: "533MHz", "667MHz" and "Auto".

3.5.7 System BIOS Cacheable

Selecting “Enabled” allows caching of the system BIOS ROM at F0000h-FFFFFh, resulting in better system performance. However, if any program writes data to this memory area, a system error may occur. The Choices are “Enabled”, and “Disabled”.

3.5.8 Video BIOS Cacheable

Selecting “Enabled” allows caching of the video BIOS, resulting in better system performance. However, if any program writes to this memory area, a system error may occur. The choices are “Enabled”, and “Dis- abled”.

3.5.9 Memory Hole At 15M-16M

Enabling this feature reserves 15 MB to 16 MB memory address space for ISA expansion cards that specifically require this setting. This makes memory from 15 MB and up unavailable to the system. Expansion cards can only access memory up to 16 MB. The default setting is “Disabled”.

3.5.10 PCI-Express Root Port Func

PCI Express Port 1/2/3/4

The default setting is “Auto.” The choices are “Enabled,” “Disabled,” and “Auto.”

PCI-E Compliancy Mode

This allows the user to select the PCI-E compliant mode. The options are [v1.0], and [v1.0a].

3.5.11 PEG / Onchip VGA Control

Use this field to select PEG or Onchip VGA. The default is AUTO.

3.5.12 On-Chip Frame Buffer Size
The On-Chip Frame Buffer Size can be set to 1 MB or 8 MB. This memory is shared with the system memory.

3.5.13 DVMT Mode
Displays the active system memory mode.

3.5.14 DVMT / FIXED Memory Size
Specify the size of DVMT / FIXED system memory to allocate for video memory.

3.5.15 Boot Display
Choose the boot display device.The default setting is "Auto" The choices are “VBIOS Default”, “CRT”, “LFP”and “LFP+CRT”.

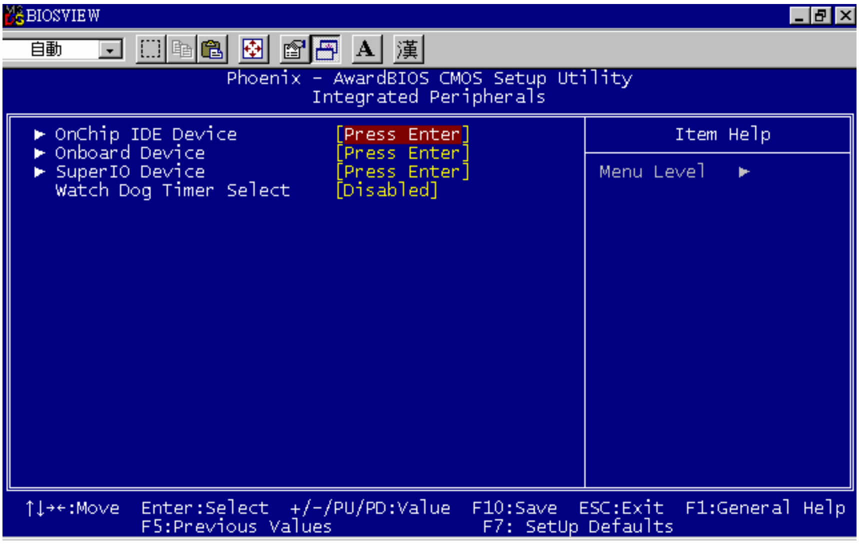
3.5.16 Panel Number
These fields allow you to select the LCD Panel type. The default values for these ports are:

- 640x480, 18bit SC
- 800x600, 18bit SC
- 1024x768, 18bit SC
- 1028x1024, 18bit DC

3.5.17 Init Display First

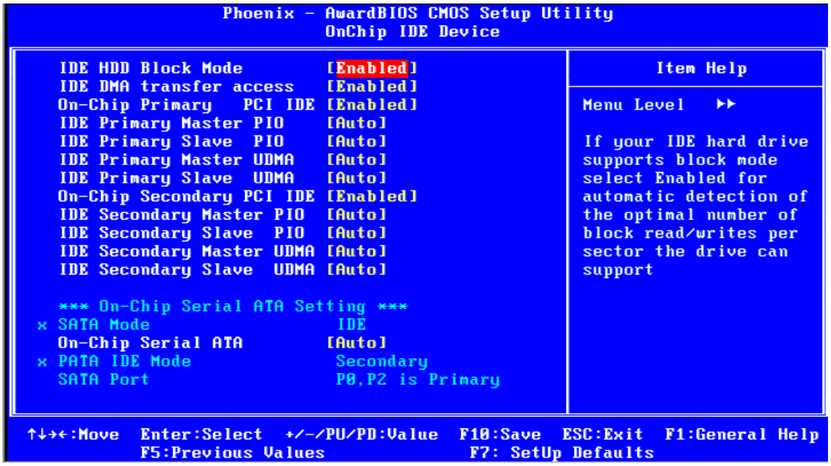
Choose the first display interface to initiate while booting. The choice is “PCI Slot” or “Onboard”.

3.6 Integrated Peripherals



```
              Phoenix - AwardBIOS CMOS Setup Utility
                     Integrated Peripherals

 ► OnChip IDE Device          [Press Enter]   |        Item Help
 ► Onboard Device             [Press Enter]   |
 ► SuperIO Device             [Press Enter]   |  Menu Level    ►
   Watch Dog Timer Select     [Disabled]      |

 ↑↓→←:Move   Enter:Select  +/-/PU/PD:Value  F10:Save  ESC:Exit  F1:General Help
             F5:Previous Values                F7: SetUp Defaults
```

Figure 3.6: Integrated peripherals



```
                Phoenix - AwardBIOS CMOS Setup Utility
                          OnChip IDE Device

    IDE HDD Block Mode          [Enabled]   |        Item Help
    IDE DMA transfer access     [Enabled]   |
    On-Chip Primary    PCI IDE  [Enabled]   | Menu Level    ▸▸
    IDE Primary Master PIO      [Auto]      |
    IDE Primary Slave  PIO      [Auto]      | If your IDE hard drive
    IDE Primary Master UDMA     [Auto]      | supports block mode
    IDE Primary Slave  UDMA     [Auto]      | select Enabled for
    On-Chip Secondary PCI IDE   [Enabled]   | automatic detection of
    IDE Secondary Master PIO    [Auto]      | the optimal number of
    IDE Secondary Slave  PIO    [Auto]      | block read/writes per
    IDE Secondary Master UDMA   [Auto]      | sector the drive can
    IDE Secondary Slave  UDMA   [Auto]      | support

    *** On-Chip Serial ATA Setting ***
  x SATA Mode                    IDE
    On-Chip Serial ATA          [Auto]
  x PATA IDE Mode                Secondary
    SATA Port                    P0,P2 is Primary

↑↓→←:Move  Enter:Select  +/-/PU/PD:Value  F10:Save  ESC:Exit  F1:General Help
           F5:Previous Values               F7: SetUp Defaults
```

Figure 3.7: On-Chip IDE Device

3.6.1 IDE HDD Block Mode

If your IDE hard drive supports block mode select Enabled for automatic detection of the optimal number of block read/writes per sector the drive can support.

3.6.2 IDE DMA Transfer Access

Use this field to enable or disable IDE DMA transfer access.

3.6.3 On-Chip Primary / Secondary IDE Device

IDE Primary Master/Slave PIO/UDMA Mode (Auto). The channel has both a master and a slave, making four IDE devices possible. Because two IDE devices may have a different Mode timing (0, 1, 2, 3, 4), it is necessary for these to be independent. The default setting "Auto" will allow auto detection to ensure optimal performance.

3.6.4 SATA Mode

The setting choices for the SATA Mode are IDE, RAID and AHCI Mode. Select [IDE] if you want to have SATA function as IDE. Select [AHCI] for Advanced Host Controller Interface (AHCI) feature, with improved SATA performance and native command queuing. Select [RAID] to use SATA for RAID.

Note: *Please refer to the PDF-format Intel(R) Matrix Storage Technology Quickstartguide and Intel(R) Matrix Storage Manager User's Manual in this CD (in the MANUAL folder) to know the necessary steps to build and configure your RAID 0, 1 system using Intel(R) Matrix Storage Technology and Matrix Storage Manager.*

3.6.5 On-Chip Serial ATA

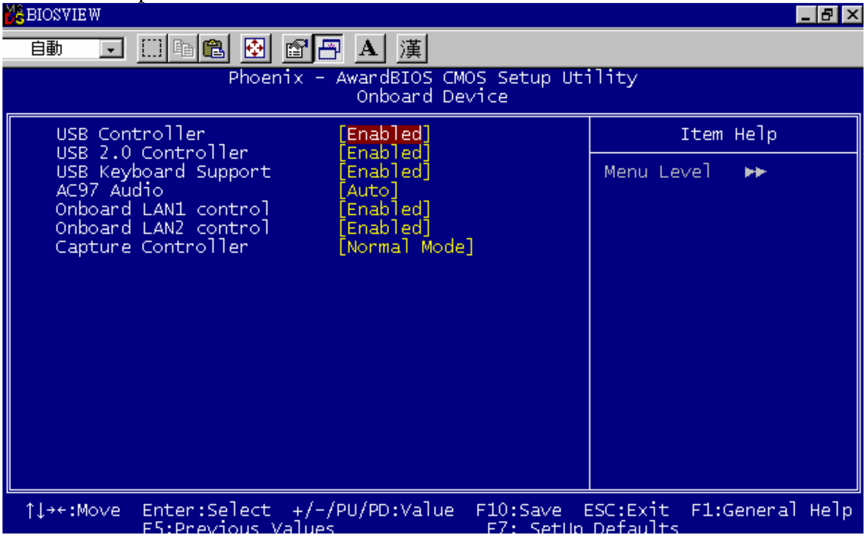
Choose the status of serial ATA. The default setting is “Auto” which lets the system arrange all parallel and serial ATA resources automatically. The “Disabled” setting disables the SATA controller. The “Combined Mode” combines PATA and SATA, and maximum of 2 IDE drives in each channel. The “Enhanced Mode” enables both SATA and PATA, and a maximum of 6 IDE drives are supported. The “SATA Only” setting means SATA is operating in legacy mode.

3.6.6 PATA IDE Mode

This item shows the parallel ATA channel. It is Secondary.

3.6.7 SATA Port

The PATA IDE mode must to be set to Secondary and SATA will display
“P0, P2 is Primary". It means SATA0 and SATA2 act as Primary Chan- nel.We have one configuration with this setting by spec. and can’t swap the channel.



```
Phoenix - AwardBIOS CMOS Setup Utility
Onboard Device

USB Controller              [Enabled]        Item Help
USB 2.0 Controller          [Enabled]
USB Keyboard Support        [Enabled]        Menu Level   ▶▶
AC97 Audio                  [Auto]
Onboard LAN1 control        [Enabled]
Onboard LAN2 control        [Enabled]
Capture Controller          [Normal Mode]

↑↓→←:Move  Enter:Select  +/-/PU/PD:Value  F10:Save  ESC:Exit  F1:General Help
F5:Previous Values  F7: SetUp Defaults
```

Figure 3.8: Onboard Device

3.6.8 USB Controller

Select Enabled if your system contains a Universal Serial Bus (USB) con- troller and you have USB peripherals. The choices are “Enabled” and
“Disabled”.

3.6.9 USB 2.0 Controller
This entry is to disable/enable the USB 2.0 controller only. The BIOS itself may/may not have high-speed USB support. If the BIOS has high speed USB support built in, the support will automatically turn on when a high speed device is attached. The choices are “Enabled” or “Disabled”.

3.6.10 USB Keyboard / Mouse Support
Select Enabled if you plan to use an USB keyboard. The choices are “Enabled” and “Disabled”.

3.6.11 AC97 Audio
Select “Disable” if you do not want to use AC-97 audio. Options are “Auto”, and “Disabled”.

3.6.12 Onboard LAN1 Control
Options are “Enabled” and “Disabled”. Select “Disabled” if you don’t want to use onboard LAN controller1.

3.6.13 Onboard LAN2 Control
Options are “Enabled” and “Disabled”. Select Disabled if you don’t want to use the onboard LAN controller2.
Note: Correct sequence of onboard Lan controllers(Lan1 & Lan2) shows in the "Onboard Device" BIOS view.

3.6.14 Capture Controller
Options are “Normal Mode” and “Enhanced Mode”. Select Enhanced Mode if you want to have more effective video capture performance. Default setting is Normal Mode.

Note: *DRAM default timings have been carefully chosen and should ONLY be changed if data is being lost. Please first contact technical sup- port.*

Super I/O Device:

3.6.15 Onboard Serial Port 1
The settings are “3F8/IRQ4”, “2F8/IRQ3”, “3E8/IRQ4”, “2E8/IRQ3”, and “Disabled” for the on-board serial connector.

3.6.16 Onboard Serial Port 2
The settings are “3F8/IRQ4”, “2F8/IRQ3”, “3E8/IRQ4”, “2E8/IRQ3”, and “Disabled” for the on-board serial connector.

3.6.17 UART Mode Select

This item allows you to select UART mode. The choices: “IrDA”, “ASKIR”, and “Normal”.

3.6.18 RxD, TxD Active

This item allows you to determine the active level of the RxD and TxD serial lines. The Choices: “Hi, Hi”, “Lo, Lo”, “Lo, Hi”, and “Hi, Lo”.

3.6.19 IR Transmission Delay

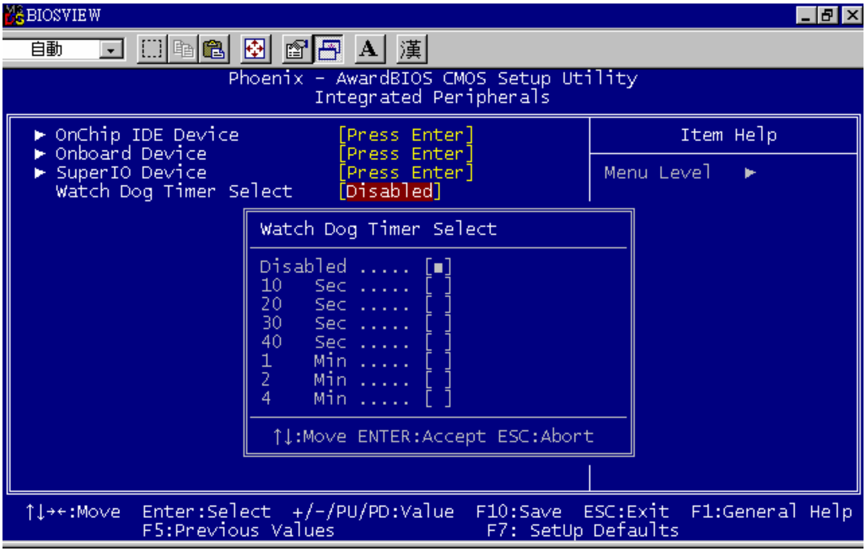
This item allows you to enable/disable IR transmission delay.
The choices are “Enabled” and “Disabled”.

3.6.20 UR2 Duplex Mode

This item allows you to select the IR half/full duplex function.
The choices are “Half” and “Full”.

3.6.21 Use IR Pins

The choices are “RxD2, TxD2” and “IR-Rx2Tx2”.



```
            Phoenix - AwardBIOS CMOS Setup Utility
                   Integrated Peripherals

 ► OnChip IDE Device          [Press Enter]         Item Help
 ► Onboard Device             [Press Enter]
 ► SuperIO Device             [Press Enter]    Menu Level   ►
   Watch Dog Timer Select     [Disabled]

               Watch Dog Timer Select

               Disabled ..... [■]
               10   Sec ..... [ ]
               20   Sec ..... [ ]
               30   Sec ..... [ ]
               40   Sec ..... [ ]
               1    Min ..... [ ]
               2    Min ..... [ ]
               4    Min ..... [ ]

                ↑↓:Move ENTER:Accept ESC:Abort

 ↑↓→←:Move  Enter:Select  +/-/PU/PD:Value  F10:Save  ESC:Exit  F1:General Help
            F5:Previous Values                F7: Setup Defaults
```

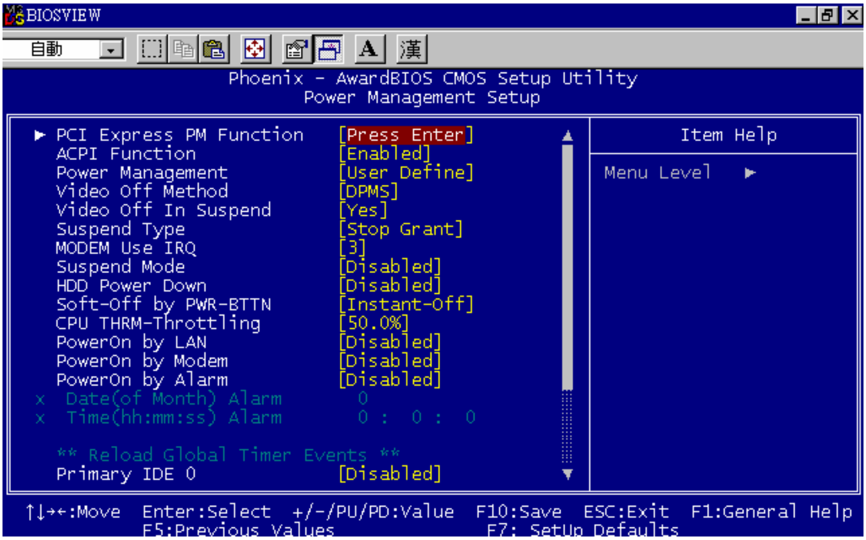
Figure 3.9: Watch Dog Timer

3.6.22 Watch Dog Timer Select

Allow User select watch Dog time or disable..

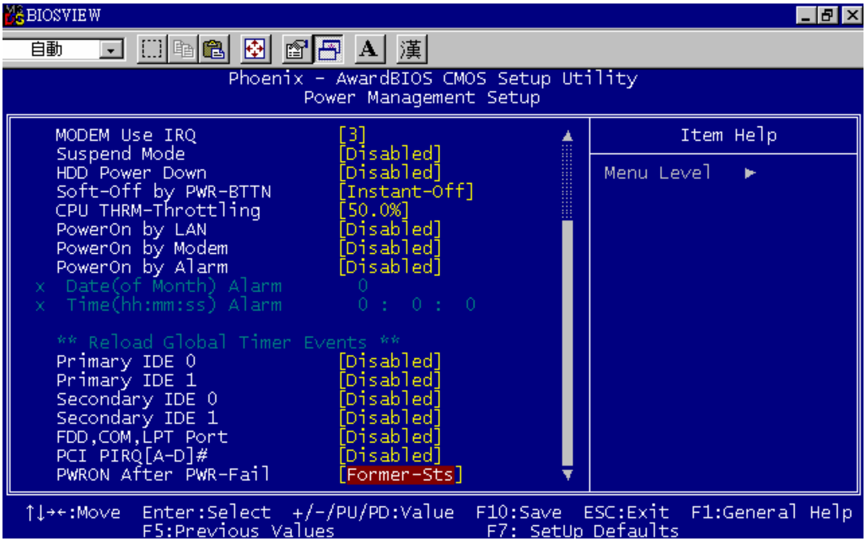
3.7 Power Management Setup

The power management setup controls the single board computer's “green” features to save power. The following screen shows the manufac- turer’s defaults.



```
                Phoenix - AwardBIOS CMOS Setup Utility
                       Power Management Setup

  ► PCI Express PM Function      [Press Enter]         Item Help
    ACPI Function                [Enabled]
    Power Management             [User Define]     Menu Level  ►
    Video Off Method             [DPMS]
    Video Off In Suspend         [Yes]
    Suspend Type                 [Stop Grant]
    MODEM Use IRQ                [3]
    Suspend Mode                 [Disabled]
    HDD Power Down               [Disabled]
    Soft-Off by PWR-BTTN         [Instant-Off]
    CPU THRM-Throttling          [50.0%]
    PowerOn by LAN               [Disabled]
    PowerOn by Modem             [Disabled]
    PowerOn by Alarm             [Disabled]
  x  Date(of Month) Alarm         0
  x  Time(hh:mm:ss) Alarm         0 :  0 :  0

    ** Reload Global Timer Events **
    Primary IDE 0                [Disabled]

 ↑↓→←:Move  Enter:Select  +/-/PU/PD:Value  F10:Save  ESC:Exit  F1:General Help
            F5:Previous Values                F7: Setup Defaults
```

Figure 3.10: Power management setup screen (a)



```
                Phoenix - AwardBIOS CMOS Setup Utility
                       Power Management Setup

    MODEM Use IRQ                [3]                   Item Help
    Suspend Mode                 [Disabled]
    HDD Power Down               [Disabled]        Menu Level  ►
    Soft-Off by PWR-BTTN         [Instant-Off]
    CPU THRM-Throttling          [50.0%]
    PowerOn by LAN               [Disabled]
    PowerOn by Modem             [Disabled]
    PowerOn by Alarm             [Disabled]
  x  Date(of Month) Alarm         0
  x  Time(hh:mm:ss) Alarm         0 :  0 :  0

    ** Reload Global Timer Events **
    Primary IDE 0                [Disabled]
    Primary IDE 1                [Disabled]
    Secondary IDE 0              [Disabled]
    Secondary IDE 1              [Disabled]
    FDD,COM,LPT Port             [Disabled]
    PCI PIRQ[A-D]#               [Disabled]
    PWRON After PWR-Fail         [Former-Sts]

 ↑↓→←:Move  Enter:Select  +/-/PU/PD:Value  F10:Save  ESC:Exit  F1:General Help
            F5:Previous Values                F7: Setup Defaults
```

Figure 3.10: Power management setup screen (b)

3.7.1 PCI Express PM Function

This allow you to control Power On by onboard LAN chip feature.

3.7.2 ACPI Function

The choices are: “Enabled” and “Disabled”.Power Management

This category allows you to select the type (or degree) of power saving and is directly related to the following modes:

- HDD Power Down
- Suspend Mode

There are three selections for Power Management, and they have fixed mode settings.

Saving Mode	Function
Min Saving	Minimum power management., Suspend Mode = 1 hr., and HDD Power Down = 15 min.
Max Saving	Maximum power management., Suspend Mode = 1 min., and HDD Power Down = 1 min.
User Defined (Default)	Allows you to set each mode individually. When not disabled, each of the ranges are from 1 min. to 1 hr. except for HDD Power Down which ranges from 1 min. to 15 min., and disabled.

3.7.3 Power Management

This item allows user to select system power saving mode.

● Min Saving Minimum power management. Suspend Mode=1 hr.

● Max Saving Maximum power management. Suspend Mode=1 min.

● User Define Allows user to set each mode individually. Suspend Mode= Disabled or 1 min ~1 hr.

3.7.4 Video Off Method

Use this to select the method to turn off the video. The choices are “Blank Screen”, “V/H SYNC+ Blank”, and “DPMS”.

3.7.5 Video Off In Suspend

When the system is in suspend mode, the video will turn off. The choices are “No” and “Yes”.

3.7.6 Suspend Type

The choices are “Stop Grant” and “PwrOn Suspend”.

3.7.7 Modem Use IRQ
This determines the IRQ that the MODEM can use.The choices are “3”, “4”, “5”, “7”, “9”, “10”, “11”, and “NA”.

3.7.8 Suspend Mode
This item allows user to determine the time of system inactivity, all devices except the CPU will be shut off.

3.7.9 HDD Power Down
This item allows user to determine the time of system inactivity, the hard disk drive will be powered down.

3.7.10 Soft-Off by PWR-BTTN
If you choose “Instant-Off”, then pushing the ATX soft power switch but- ton once will switch the system to “system off” power mode. You can choose “Delay 4 sec”. If you do, then pushing the button for more than 4 seconds will turn off the system, whereas pushing the button momentarily
(for less than 4 seconds) will switch the system to “suspend” mode.

3.7.11 PowerOn by LAN
This item allows you to power on the system by LAN. The choices are “Enabled” and “Disabled”.

3.7.12 CPU THRM-Throttling
This option controlls the CPU speed as a percentage of regular power. the choices are 87.5%, 75%, 62.5%, 50%, 37.5%, 25% .12% and 12.5%.

3.7.13 PowerOn by Modem
To enabled or disable the function to power on the system via a Modem connection from a remote host. The choice “Enabled” and “Disabled”.

3.7.14 PowerOn by Alarm
The choices are “Enabled” and “Disabled”. Fields that follow below indicate date of current month and time of alarm settings, if enabled.

3.7.15 Primary IDE 0 (1) and Secondary IDE 0 (1)
When Enabled, the system will resume from suspend mode if Primary IDE 0 (1) or Secondary IDE 0 (1) becomes active. The choices are “Enabled” and “Disabled”.

3.7.16 FDD, COM, LPT PORT
When Enabled, the system will resume from suspend mode if the

FDD, interface, COM port, or LPT port is active. The choices are “Enabled” and “Disabled”.

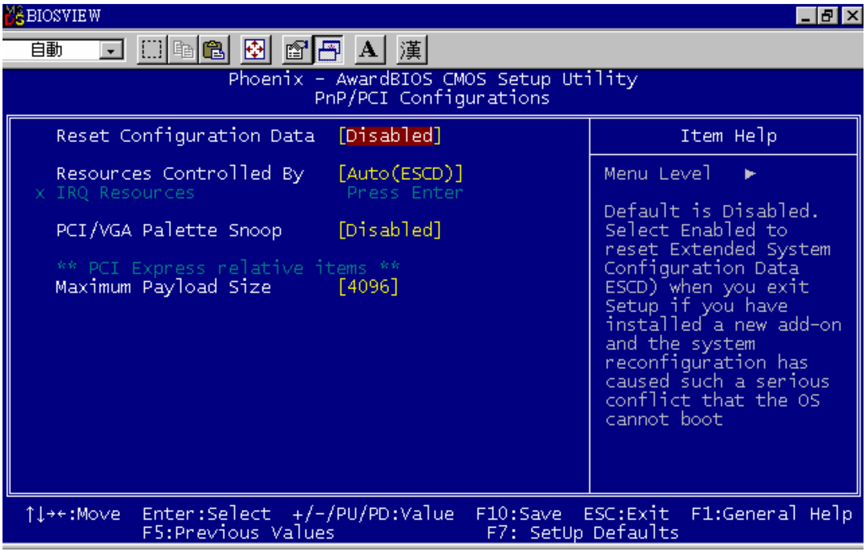
3.7.17 PCI PIRQ [A-D]#

When Enabled, the system resumes from suspend mode if an interrupt occurs. The choices are “Enabled” and “Disabled”.

3.7.18 PWRON After PWR-Fail

Use this to set up the system after power failure. The “Off” setting keeps the system powered off after power failure, the “On” setting boots up the system after failure, and the “Former-Sts” returns the system to the status before power failure.

3.8 PnP/PCI Configurations



```
           Phoenix - AwardBIOS CMOS Setup Utility
                   PnP/PCI Configurations

   Reset Configuration Data      [Disabled]        Item Help

   Resources Controlled By       [Auto(ESCD)]    Menu Level   ►
 x IRQ Resources                  Press Enter
                                                 Default is Disabled.
   PCI/VGA Palette Snoop         [Disabled]      Select Enabled to
                                                 reset Extended System
   ** PCI Express relative items **              Configuration Data
   Maximum Payload Size          [4096]          ESCD) when you exit
                                                 Setup if you have
                                                 installed a new add-on
                                                 and the system
                                                 reconfiguration has
                                                 caused such a serious
                                                 conflict that the OS
                                                 cannot boot

 ↑↓→←:Move  Enter:Select  +/-/PU/PD:Value  F10:Save  ESC:Exit  F1:General Help
            F5:Previous Values               F7: SetUp Defaults
```

Figure 3.11: PnP/PCI configurations screen

3.8.1 Reset Configuration Data

The default is Disabled. Select Enabled to reset Extended System Configuration Data (ESCD) if you have installed a new add-on card, and system configuration is in such a state that the OS cannot boot.

3.8.2 Resources Controlled By

The commands here are “Auto(ESCD)” or “Manual”. Choosing “Manual” requires you to choose resources from the following sub-menu. “Auto(ESCD)” automatically configures all of the boot and Plug and Play devices, but you must be using Windows 95 or above.

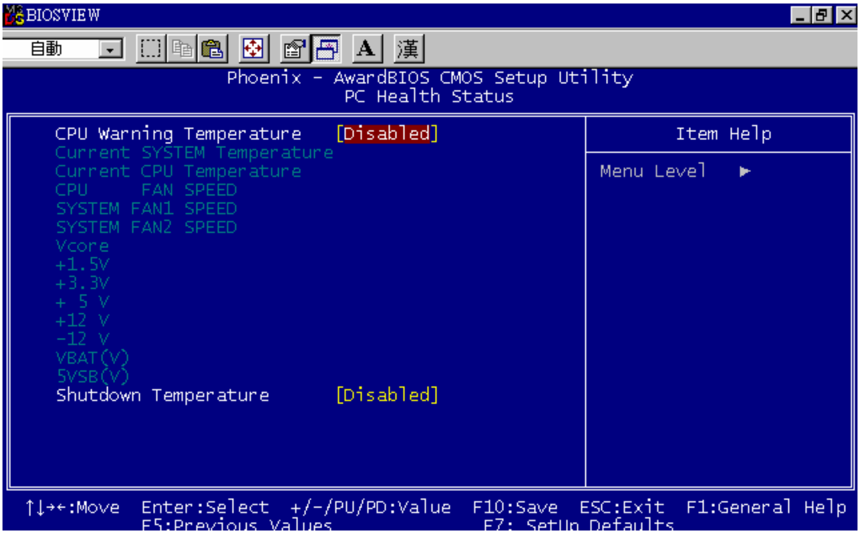
3.8.3 PCI / VGA Palette Snoop

This is set to “Disabled” by default.

3.8.4 Maximum Payload Size

This allows you to set the maximum TLP payload size for PCI Express devices. The options are [128 bytes], [256 bytes], [512 bytes], [1024 bytes], [2048 bytes], and [4096 bytes].

3.9 PC Health Status



```
               Phoenix - AwardBIOS CMOS Setup Utility
                         PC Health Status

   CPU Warning Temperature     [Disabled]          Item Help
   Current SYSTEM Temperature
   Current CPU Temperature                      Menu Level   ►
   CPU     FAN  SPEED
   SYSTEM FAN1 SPEED
   SYSTEM FAN2 SPEED
   Vcore
   +1.5V
   +3.3V
   + 5 V
   +12 V
   -12 V
   VBAT(V)
   5VSB(V)
   Shutdown Temperature        [Disabled]

 ↑↓→←:Move  Enter:Select  +/-/PU/PD:Value  F10:Save  ESC:Exit  F1:General Help
            F5:Previous Values               F7: Setup Defaults
```

Figure 3.12: PC Health Status Screen

3.9.1 CPU Warning Temperature

This item will prevent the CPU from overheating. The choices are "Disabled", "60C/140F", "63C/145F", "66C/151F", "70C/158F", "75C/167F", "80C/176F", "85C/185F", "90C/194F", and "95C/205F".

3.9.2 Current System Temperature

This shows you the current temperature of system.

3.9.3 Current CPU Temperature

This shows the current CPU temperature.

3.9.4 CPU FAN Speed

This shows the current CPU FAN operating speed.

3.9.5 System FAN 1 / 2 Speed

This shows the current System FAN operating speed.

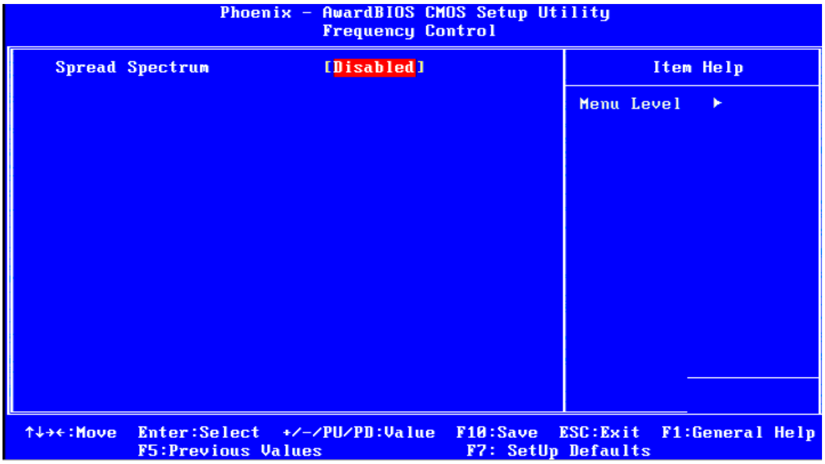
3.9.6 VCORE and Other Voltages

This shows the voltage of VCORE, +1.5V, +3.3, +5V, +12V, -12V, -5V, VBAT(V), and 5VSB(V).

3.9.7 Shutdown Temperature

This item enables users to set the limitation of CPU temperature, the range is from 85°C through 100°C.

3.10 Frequency / Voltage Control



```
              Phoenix - AwardBIOS CMOS Setup Utility
                        Frequency Control

   Spread Spectrum              [Disabled]        Item Help

                                                  Menu Level   ►

 ↑↓→←:Move  Enter:Select  +/-/PU/PD:Value  F10:Save  ESC:Exit  F1:General Help
            F5:Previous Values               F7: SetUp Defaults
```

Figure 3.14: Spread Spectrum Control screen

3.10.1 Spread Spectrum

This setting allows you to reduce EMI by modulating the signals the CPU generates so that the spikes are reduced to flatter curves. This is achieved by varying the frequency slightly so that the signal does not use any particular frequency for more than a moment. The choices are “Disabled” and “Enabled”.

3.11 Password Setting

Follow these steps to change the password.

1. Choose the “Set Password” option from the “Initial Setup Screen” menu and press <Enter>. The screen displays the following message:

[redacted]

2. Press <Enter>.
3. If the CMOS is good and this option has been used to change the default password, the user is asked for the password stored in the CMOS. The screen displays the following message:

[redacted]

4. Type the current password and press <Enter>.
5. After pressing <Enter> (ROM password) or the current password (user-defined), you can change the password stored in the CMOS. The password must be no longer than eight (8) characters.

Remember, to enable the password setting feature, you must first select either “Setup” or “System” from the “Advanced BIOS Features” menu.

3.12 Save & Exit Setup

If you select this and press <Enter>, the values entered in the setup utilities will be recorded in the CMOS memory of the chipset. The processor will check this every time you turn your system on and compare this to what it finds as it checks the system. This record is required for the system to operate.

3.13 Exit Without Saving

Selecting this option and pressing <Enter> lets you exit the setup program without recording any new values or changing old ones.

4

Chipset Installation

Chapter 4 Chipset Installation

4.1 Before you begin

To facilitate the installation of the enhanced display drivers and utility software, read the instructions in this chapter carefully. The drivers for the DVS-510 are located on the software installation CD. The Intel® Chipset Software Installation Utility is not required on any systems running Windows NT 4.0. Updates are provided via Service Packs from Microsoft*.

Note: *The files on the software installation CD are compressed. Do not attempt to install the driv- ers by copying the files manually. You must use the supplied SETUP program to install the driv- ers.*

Before you begin, it is important to note that most display drivers need to have the relevant software application already installed in the system prior to installing the enhanced display drivers. In addition, many of the installation procedures assume that you are familiar with both the relevant software applications and operating system commands. Review the relevant operating system commands and the pertinent sections of your application software's user manual before performing the installation.

4.2 Introduction

The Intel® Chipset Software Installation (CSI) utility installs the Windows INF files that outline to the operating system how the chipset components will be configured. This is needed for the proper functioning of the following features:

- Core PCI PnP services
- IDE Ultra ATA 100/66/33 and Serial ATA interface support
- USB 1.1/2.0 support (USB 2.0 driver needs to be installed separately for Win98)
- Identification of Intel® chipset components in the Device Manager

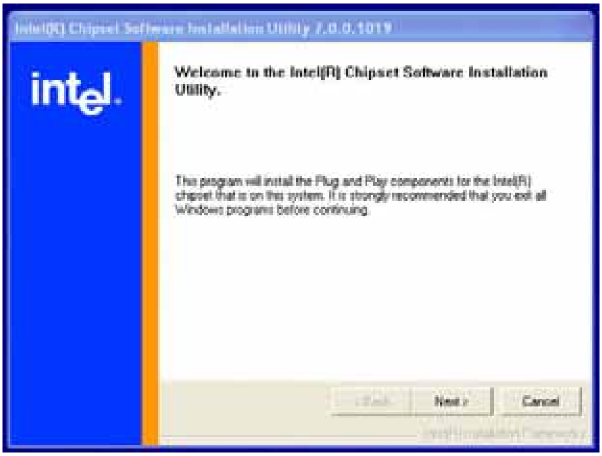
- Integrates superior video features. These include filtered sealing of 720 pixel DVD content, and MPEG-2 motion compensation for software DVD

Note: *This utility is used for the following versions of Windows system, and it has to be installed before installing all the other drivers:*

- Windows 2000
- Windows XP

4.3 Windows XP Driver Setup

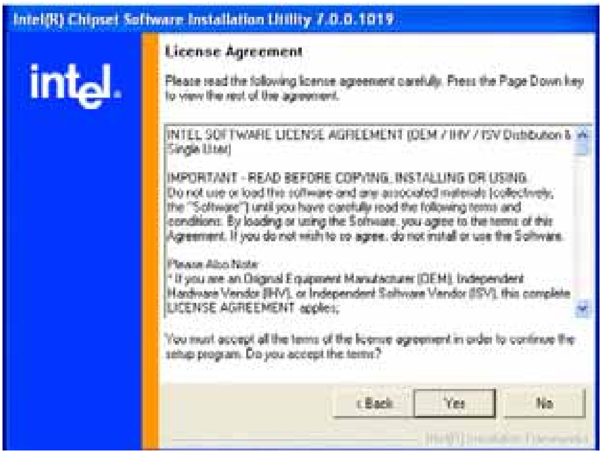
1. Insert the driver CD into your system's CD-ROM drive. All the drivers of DVS-510 are under DVS-510_CD\01_DVS-510_Driver.
2. For the driver of Intel 945GM chipset is under DVS-510_CD\01_DVS-510_Driver\01_945 chipset. Double click the icon which is in the folder to install driver. To take Windows XP as example.
3. Click "Next" when you see the following message.


Intel(R) Chipset Software Installation Utility 7.0.0.1019

Welcome to the Intel(R) Chipset Software Installation Utility.

This program will install the Plug and Play components for the Intel(R) chipset that is on this system. It is strongly recommended that you exit all Windows programs before continuing.

4. Click "Yes" when you see the following message.


Intel(R) Chipset Software Installation Utility 7.0.0.1019

License Agreement

Please read the following license agreement carefully. Press the Page Down key to view the rest of the agreement.

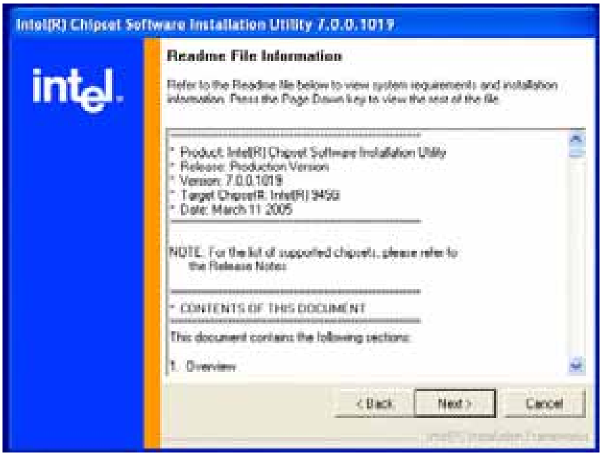
INTEL SOFTWARE LICENSE AGREEMENT (OEM / IHV / ISV Distribution & Single User)

IMPORTANT - READ BEFORE COPYING, INSTALLING OR USING.
Do not use or load this software and any associated materials (collectively, the "Software") until you have carefully read the following terms and conditions. By loading or using the Software, you agree to the terms of this Agreement. If you do not wish to so agree, do not install or use the Software.

Please Also Note:
* If you are an Original Equipment Manufacturer (OEM), Independent Hardware Vendor (IHV), or Independent Software Vendor (ISV), this complete LICENSE AGREEMENT applies;

You must accept all the terms of the license agreement in order to continue the setup program. Do you accept the terms?

5. Click "Next" when you see the following message.


Intel(R) Chipset Software Installation Utility 7.0.0.1019

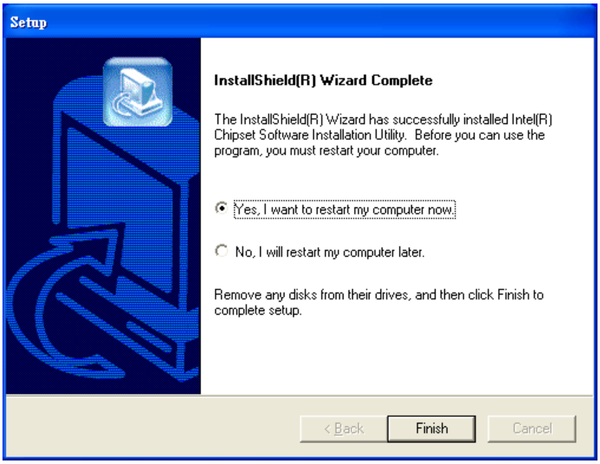
Readme File Information

Refer to the Readme file below to view system requirements and installation information. Press the Page Down key to view the rest of the file.

```
*  Product: Intel(R) Chipset Software Installation Utility
*  Release: Production Version
*  Version: 7.0.0.1019
*  Target Chipset#: Intel(R) 945G
*  Date: March 11 2005

NOTE:  For the list of supported chipsets, please refer to
    the Release Notes

*  CONTENTS OF THIS DOCUMENT

This document contains the following sections:

1.  Overview
```

6. When the following message appears, click "Finish" to complete

the installation and restart Windows.



InstallShield(R) Wizard Complete

The InstallShield(R) Wizard has successfully installed Intel(R) Chipset Software Installation Utility. Before you can use the program, you must restart your computer.

[✓ Yes, I want to restart my computer now.]

No, I will restart my computer later.

Remove any disks from their drives, and then click Finish to complete setup.

5

VGA Setup

Chapter 5 VGA Setup

5.1 Introduction

The Intel 945GM integrated graphics controller provides an analog display port, LVDS and TV-out interface. You need to install the VGA driver to enable the function.

The Intel 945GM integrated graphics controller includes the following features.

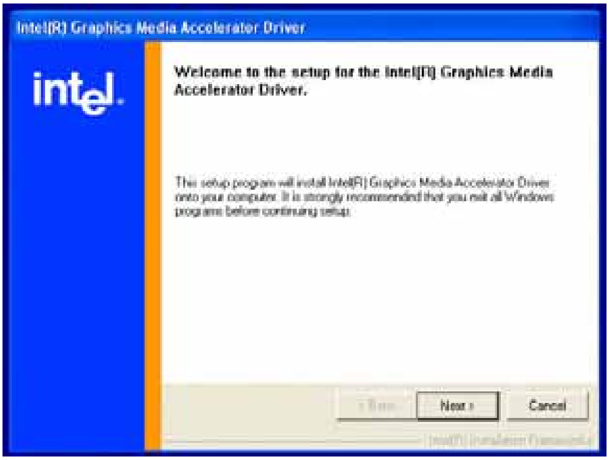
- **Intel Graphics Media Accelerator 950**: Incorporating the latest Microsoft* DirectX*9 support capabilities, it allows software developers to create lifelike environments and characters. Dual independent display, enhanced display modes for widescreen flat panels, and optimized 3D support deliver an intense and realistic visual experience without requiring a separate graphics card.
- **Intel Dynamic Video Memory Technology (DVMT3.0)**: DVS-510 handles diverse applications by providing the maximum availability of system memory for general computer usage, while supplying additional graphics memory when a 3D-intensive application requests it. The amount of video memory is dependent upon the amount of pre-allocated memory set for your system plus something called Dynamic Video Memory Technology (DVMT).
- **LVDS Interface**: DVS-510 provide 18-bit dual channel LVDS interface supporting up to WUXGA(1600X1200) panel resolution.
- **TV-Out**: DVS-510 Supports PAL/ NTSC TV systems

5.2 Windows XP Driver Setup

Insert the driver CD into your system's CD-ROM drive and find Graphic driver under DVS-510_CD\01_DVS-510_Driver\ 02_Graphics.

The following installation procedure is for Windows XP. For other operating systems please do a manual installation.

1. Click "Next" to continue the installation.


Intel(R) Graphics Media Accelerator Driver
intel.

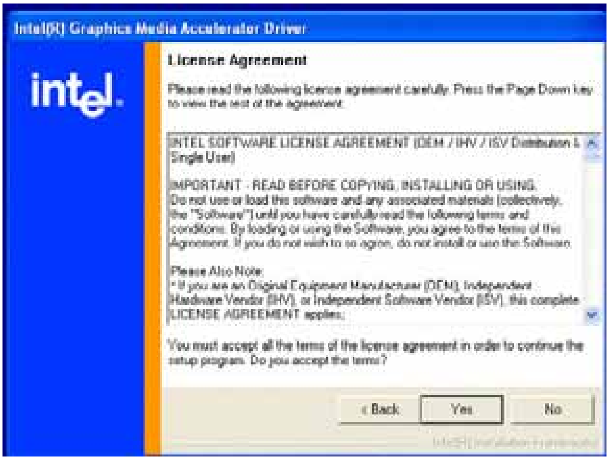
Welcome to the setup for the Intel(R) Graphics Media Accelerator Driver.

This setup program will install Intel(R) Graphics Media Accelerator Driver onto your computer. It is strongly recommended that you exit all Windows programs before continuing setup.

< Back
Next >
Cancel
Intel(R) Installation Frameworks

2. You will see a welcome window. Please click "Yes" to continue the installation.

.



License Agreement

Please read the following license agreement carefully. Press the Page Down key to view the rest of the agreement.

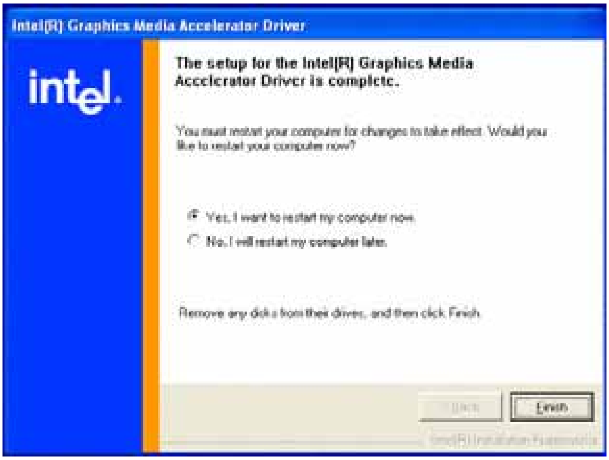
INTEL SOFTWARE LICENSE AGREEMENT (OEM / IHV / ISV Distribution & Single User)

IMPORTANT - READ BEFORE COPYING, INSTALLING OR USING.
Do not use or load this software and any associated materials (collectively, the "Software") until you have carefully read the following terms and conditions. By loading or using the Software, you agree to the terms of this Agreement. If you do not wish to so agree, do not install or use the Software.

Please Also Note:
* If you are an Original Equipment Manufacturer (OEM), Independent Hardware Vendor (IHV), or Independent Software Vendor (ISV), this complete LICENSE AGREEMENT applies;

You must accept all the terms of the license agreement in order to continue the setup program. Do you accept the terms?

3. Click “Finish” to complete the installation and restart the computer now or later.


Intel(R) Graphics Media Accelerator Driver

The setup for the Intel(R) Graphics Media Accelerator Driver is complete.

You must restart your computer for changes to take effect. Would you like to restart your computer now?

[✓ Yes, I want to restart my computer now.]

No, I will restart my computer later.

Remove any disks from their drives, and then click Finish.

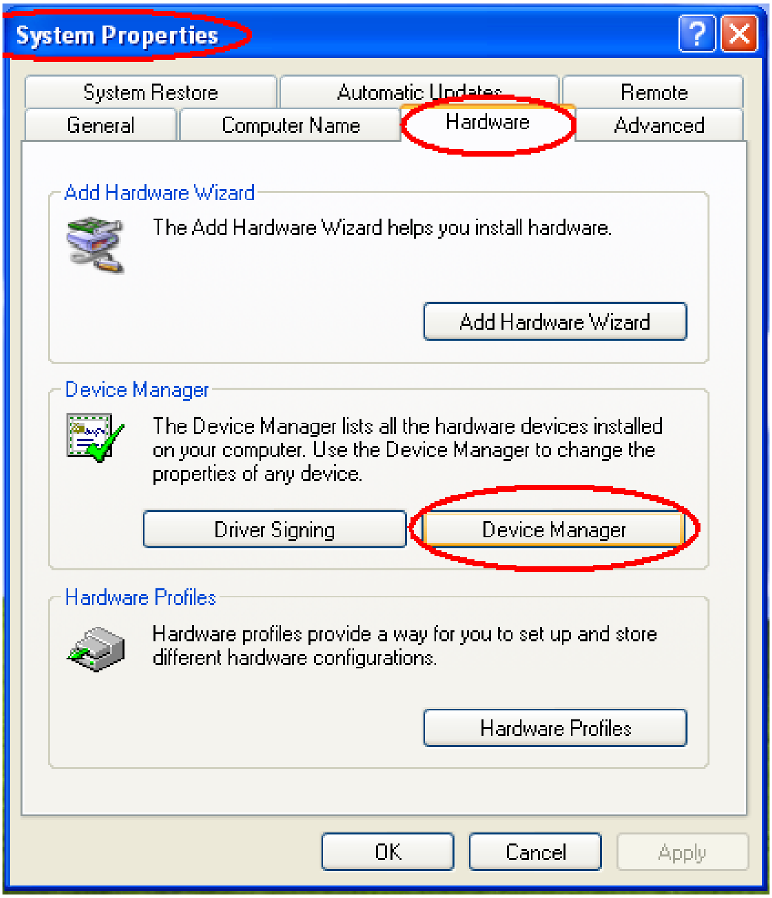
6

Video capture installation

Chapter 6 Video capture installation

6.1 Driver installation of video capture chip

Step 1: Pop-up the “System Properties” window, choose the “Hardware” page, and press the “Device Manager” bottom.

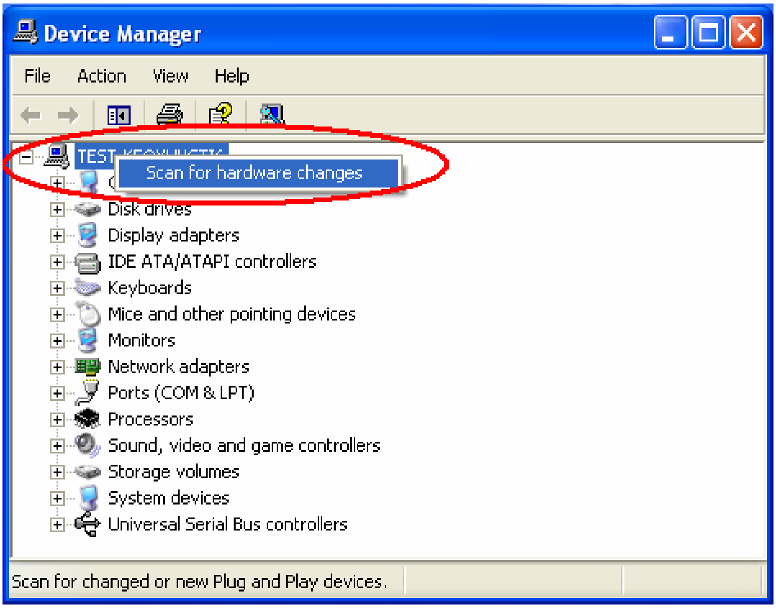


The Add Hardware Wizard helps you install hardware.

The Device Manager lists all the hardware devices installed on your computer. Use the Device Manager to change the properties of any device.

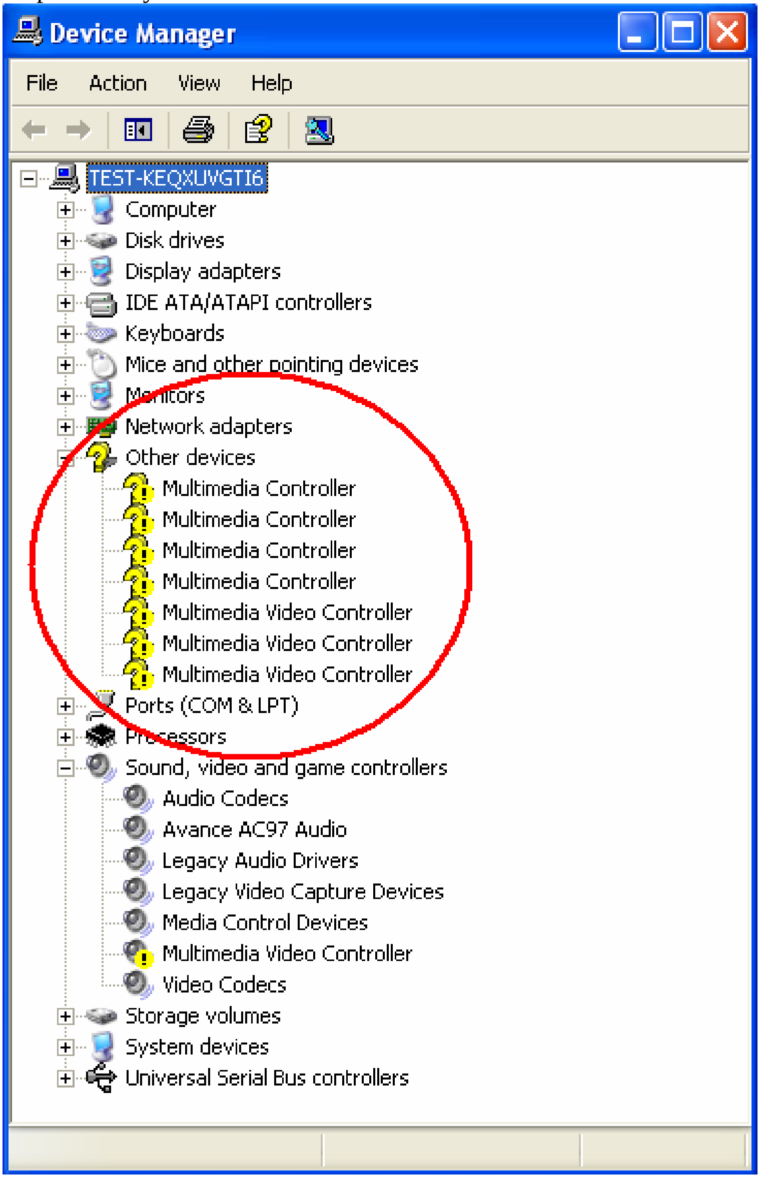
Hardware profiles provide a way for you to set up and store different hardware configurations.

Step 2: Click the PC icon and press the left bottom of the mouse. Press the "Scan for hardware changes".



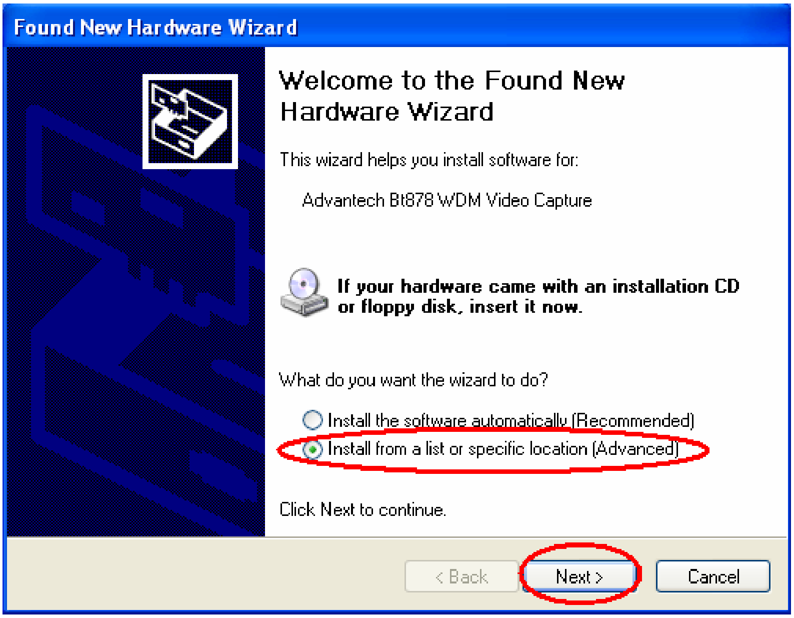
Scan for changed or new Plug and Play devices.

Step 3: The system will show the un-known devices like below window.



TEST-KEQXUVGTI6
Computer
Disk drives
Display adapters
IDE ATA/ATAPI controllers
Keyboards
Mice and other pointing devices
Monitors
Network adapters
Other devices
Multimedia Controller
Multimedia Controller
Multimedia Controller
Multimedia Controller
Multimedia Video Controller
Multimedia Video Controller
Multimedia Video Controller
Ports (COM & LPT)
Processors
Sound, video and game controllers
Audio Codecs
Avance AC97 Audio
Legacy Audio Drivers
Legacy Video Capture Devices
Media Control Devices
Multimedia Video Controller
Video Codecs
Storage volumes
System devices
Universal Serial Bus controllers

Step 4: Click the below icon to specify the driver location.



Found New Hardware Wizard

Welcome to the Found New Hardware Wizard

This wizard helps you install software for:

Advantech Bt878 WDM Video Capture

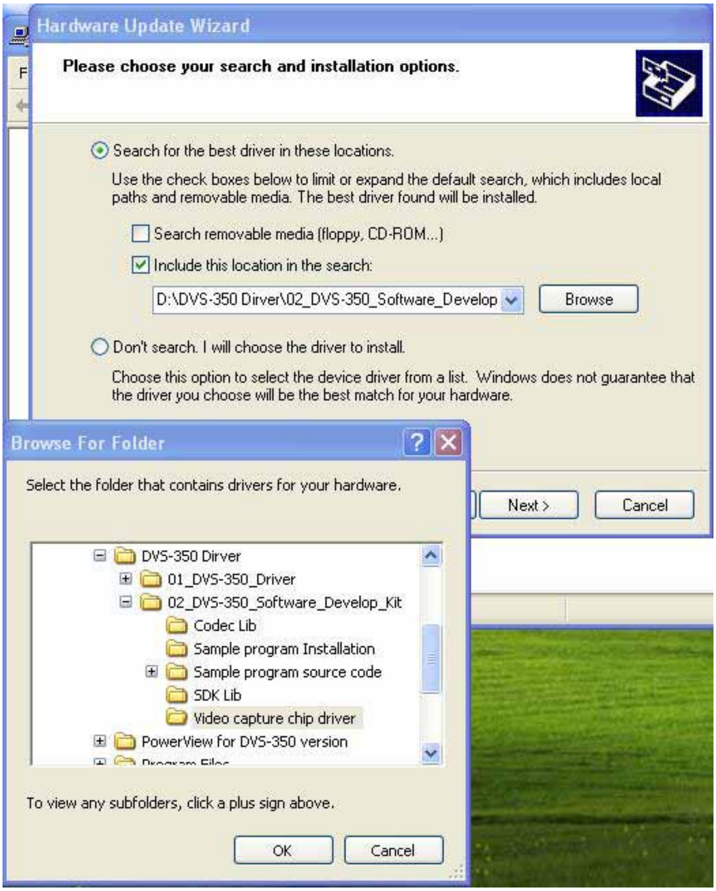
If your hardware came with an installation CD or floppy disk, insert it now.

What do you want the wizard to do?

- ○ Install the software automatically (Recommended)
- ● Install from a list or specific location (Advanced)

Click Next to continue.

Step 5: Specify the driver under the DVS-510_CD\01_DVS-510_Driver\05_BT878 Driver


Hardware Update Wizard

Please choose your search and installation options.

[✓ Search for the best driver in these locations.]

Use the check boxes below to limit or expand the default search, which includes local paths and removable media. The best driver found will be installed.

☐ Search removable media (floppy, CD-ROM...)

☑ Include this location in the search:

D:\DVS-350 Dirver\02_DVS-350_Software_Develop

Don't search. I will choose the driver to install.

Choose this option to select the device driver from a list. Windows does not guarantee that the driver you choose will be the best match for your hardware.

Browse For Folder

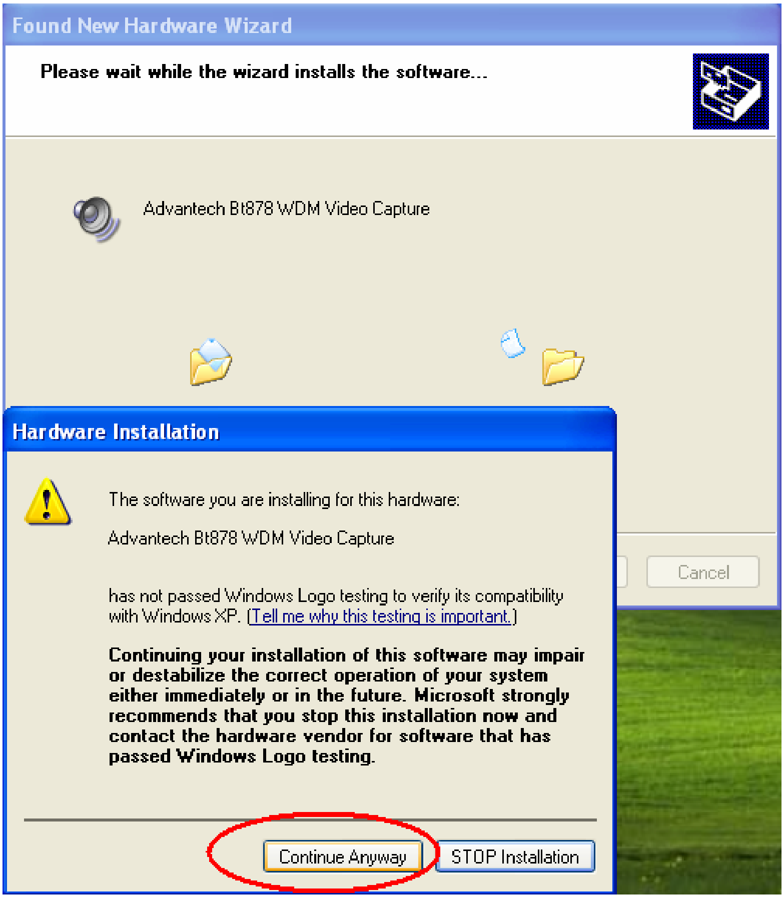
Select the folder that contains drivers for your hardware.

```
⊟ DVS-350 Dirver
    ⊞ 01_DVS-350_Driver
    ⊟ 02_DVS-350_Software_Develop_Kit
        Codec Lib
        Sample program Installation
      ⊞ Sample program source code
        SDK Lib
        Video capture chip driver
⊞ PowerView for DVS-350 version
⊞ Program Files
```

To view any subfolders, click a plus sign above.

Step 6: Push the “Next” bottom to process the installation.

Step 7: Continuing the installation.


Found New Hardware Wizard

Please wait while the wizard installs the software...

Advantech Bt878 WDM Video Capture

Hardware Installation

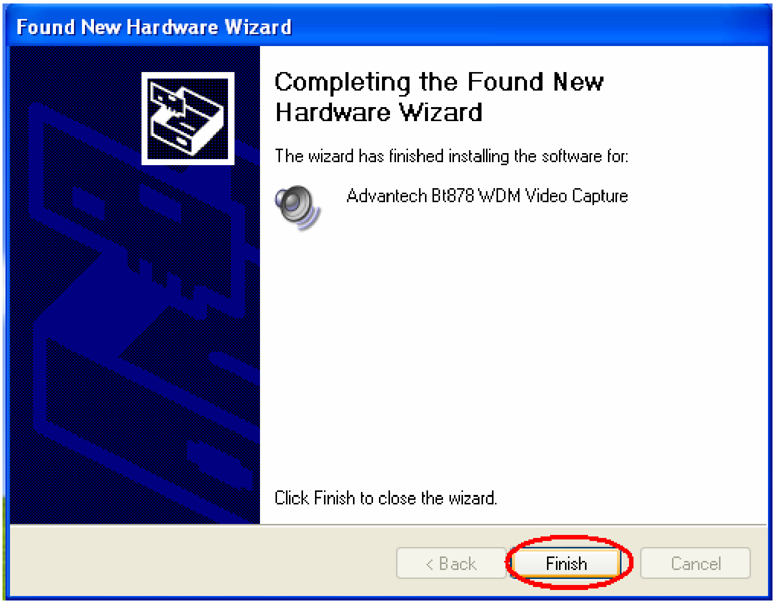
The software you are installing for this hardware:

Advantech Bt878 WDM Video Capture

has not passed Windows Logo testing to verify its compatibility with Windows XP. (Tell me why this testing is important.)

Continuing your installation of this software may impair or destabilize the correct operation of your system either immediately or in the future. Microsoft strongly recommends that you stop this installation now and contact the hardware vendor for software that has passed Windows Logo testing.

Step 8: Press the “Finish” bottom to finish the first circle installation. Then repeat the installation step 1~8 until all the un-known devices are all installed.


Found New Hardware Wizard

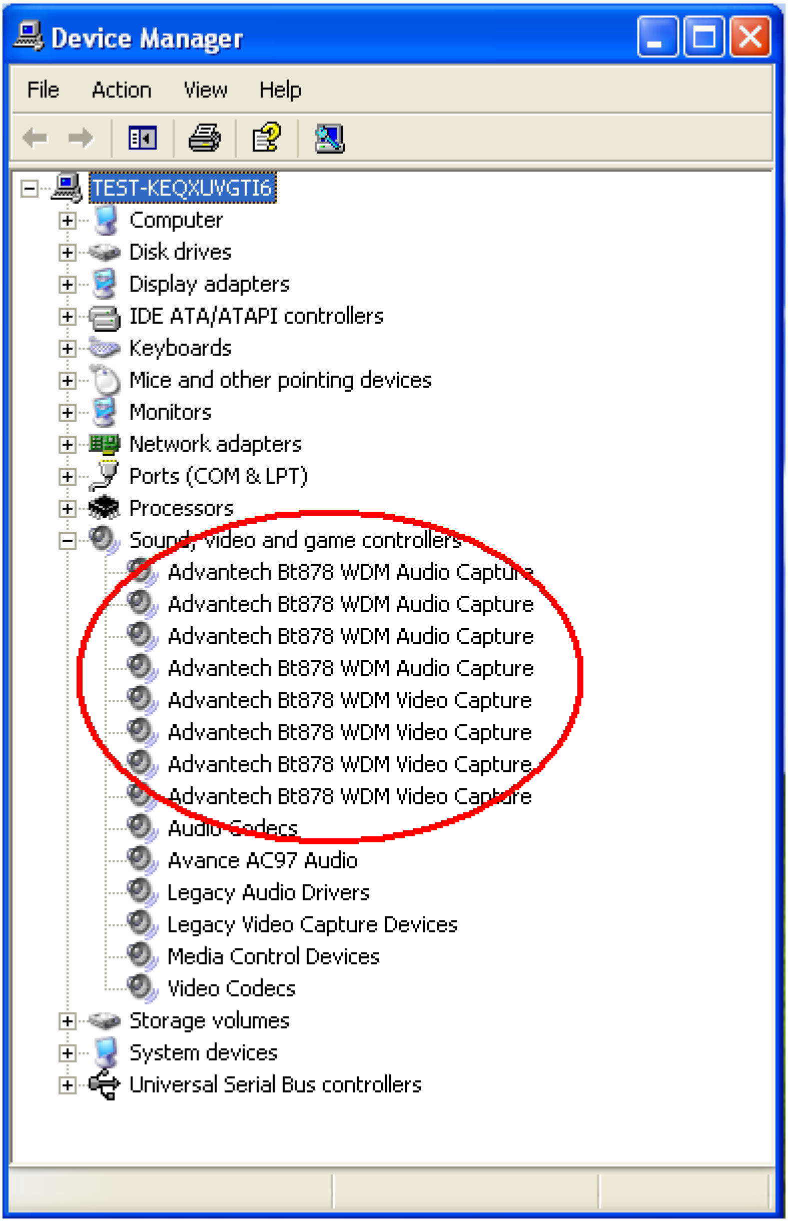
Completing the Found New Hardware Wizard

The wizard has finished installing the software for:

Advantech Bt878 WDM Video Capture

Click Finish to close the wizard.

Step 9: From below window, we know there are 8 new items are installed.



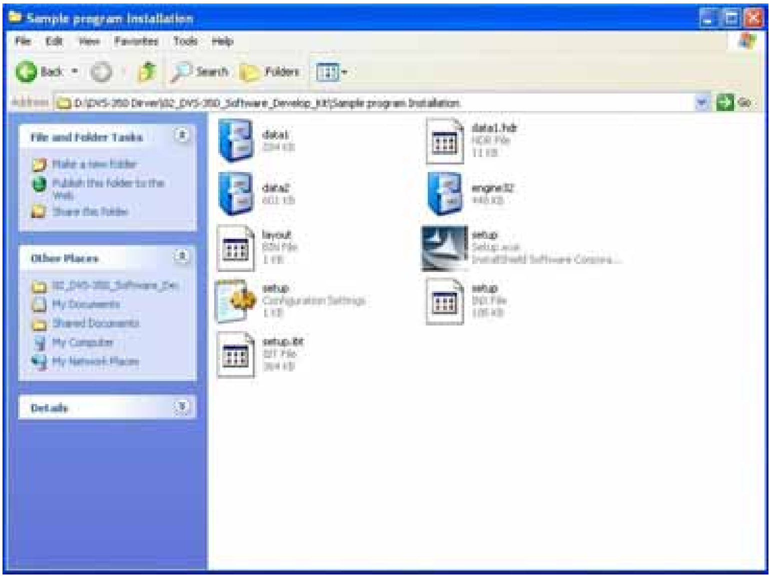
Device Manager

File Action View Help

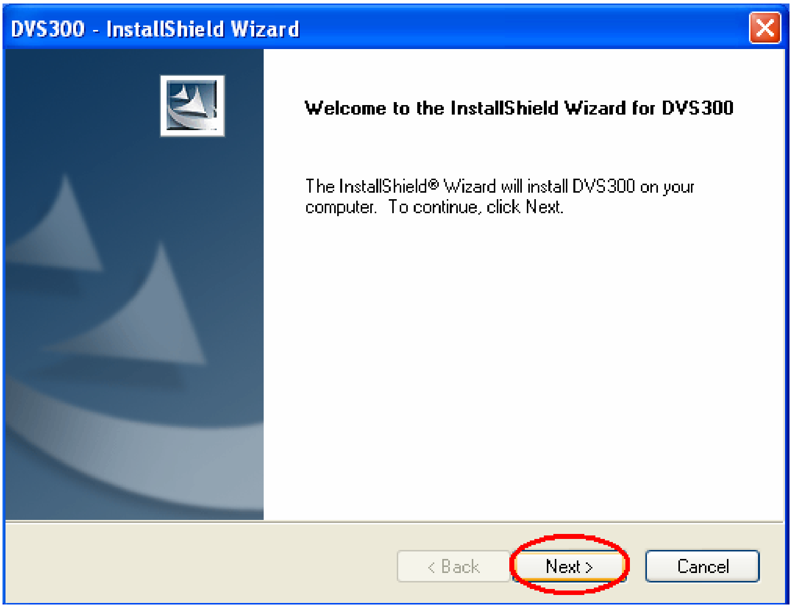
TEST-KEQXUVGTI6
Computer
Disk drives
Display adapters
IDE ATA/ATAPI controllers
Keyboards
Mice and other pointing devices
Monitors
Network adapters
Ports (COM & LPT)
Processors
Sound, video and game controllers
Advantech Bt878 WDM Audio Capture
Advantech Bt878 WDM Audio Capture
Advantech Bt878 WDM Audio Capture
Advantech Bt878 WDM Audio Capture
Advantech Bt878 WDM Video Capture
Advantech Bt878 WDM Video Capture
Advantech Bt878 WDM Video Capture
Advantech Bt878 WDM Video Capture
Audio Codecs
Avance AC97 Audio
Legacy Audio Drivers
Legacy Video Capture Devices
Media Control Devices
Video Codecs
Storage volumes
System devices
Universal Serial Bus controllers

6.2 Installation of DVS-510 Demo Program

Step 1: Install the DVS-510 demo program. The executive file is in the path: DVS-510_CD\02_DVS-510_Software_Develop_Kit \Demo Program



Step 2: Press the "Next" bottom to begin the installation.

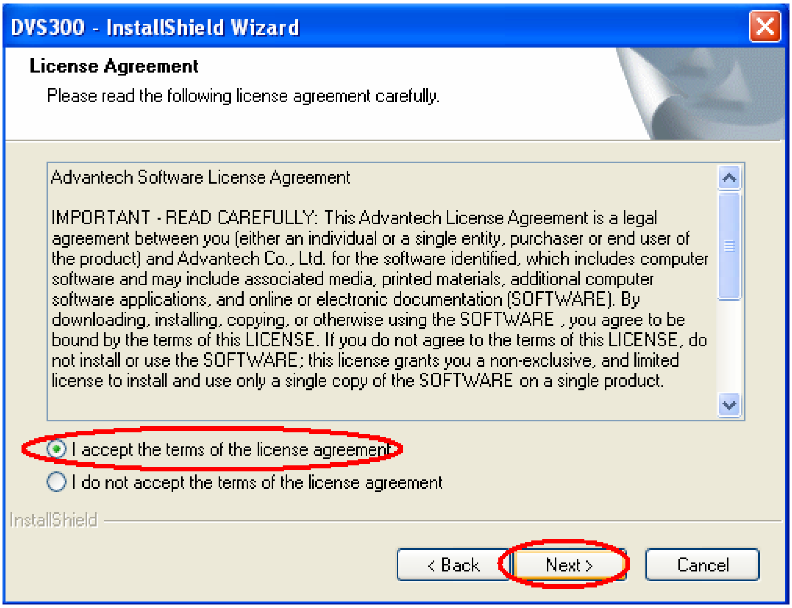


DVS300 - InstallShield Wizard

Welcome to the InstallShield Wizard for DVS300

The InstallShield® Wizard will install DVS300 on your computer. To continue, click Next.

Step 3: Accept the license agreement and continue the installation.



DVS300 - InstallShield Wizard

License Agreement

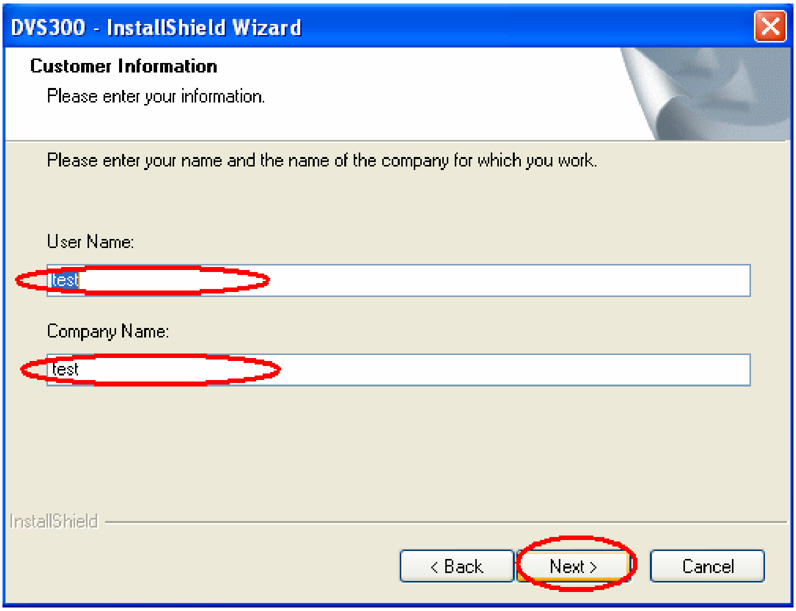
Please read the following license agreement carefully.

Advantech Software License Agreement

IMPORTANT - READ CAREFULLY: This Advantech License Agreement is a legal agreement between you (either an individual or a single entity, purchaser or end user of the product) and Advantech Co., Ltd. for the software identified, which includes computer software and may include associated media, printed materials, additional computer software applications, and online or electronic documentation (SOFTWARE). By downloading, installing, copying, or otherwise using the SOFTWARE , you agree to be bound by the terms of this LICENSE. If you do not agree to the terms of this LICENSE, do not install or use the SOFTWARE; this license grants you a non-exclusive, and limited license to install and use only a single copy of the SOFTWARE on a single product.

[✓ I accept the terms of the license agreement]
I do not accept the terms of the license agreement

Step 4: Key in your name and company name. Then press the “Next” bottom to continue.


DVS300 - InstallShield Wizard

Customer Information

Please enter your information.

Please enter your name and the name of the company for which you work.

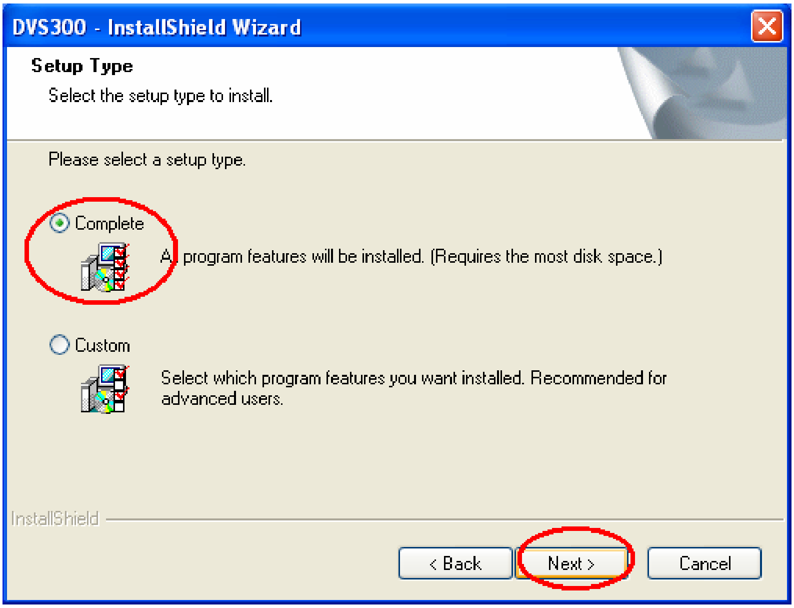
User Name:

test

Company Name:

test

Step 5: Choose the setup type you want and next.



DVS300 - InstallShield Wizard

Setup Type

Select the setup type to install.

Please select a setup type.

[✓ Complete]

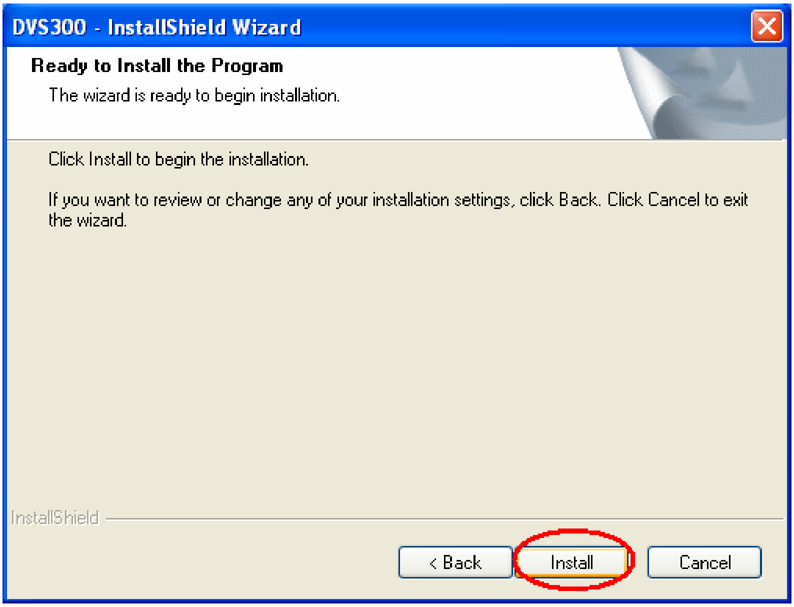
All program features will be installed. (Requires the most disk space.)

Custom

Select which program features you want installed. Recommended for advanced users.

InstallShield

Step 6: Beginning the installation.



DVS300 - InstallShield Wizard

Ready to Install the Program

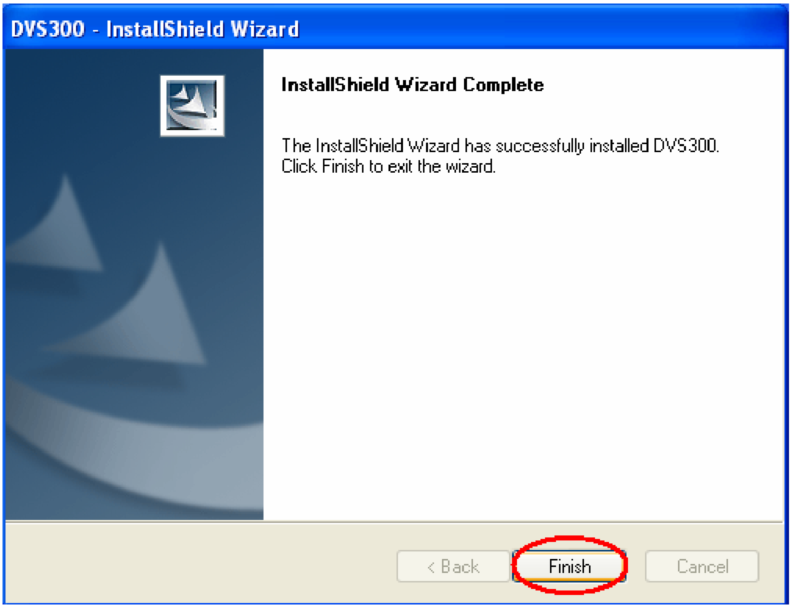
The wizard is ready to begin installation.

Click Install to begin the installation.

If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.

InstallShield

Step 7: Finished the installation of DVS-350 demo program.



DVS300 - InstallShield Wizard

InstallShield Wizard Complete

The InstallShield Wizard has successfully installed DVS300. Click Finish to exit the wizard.

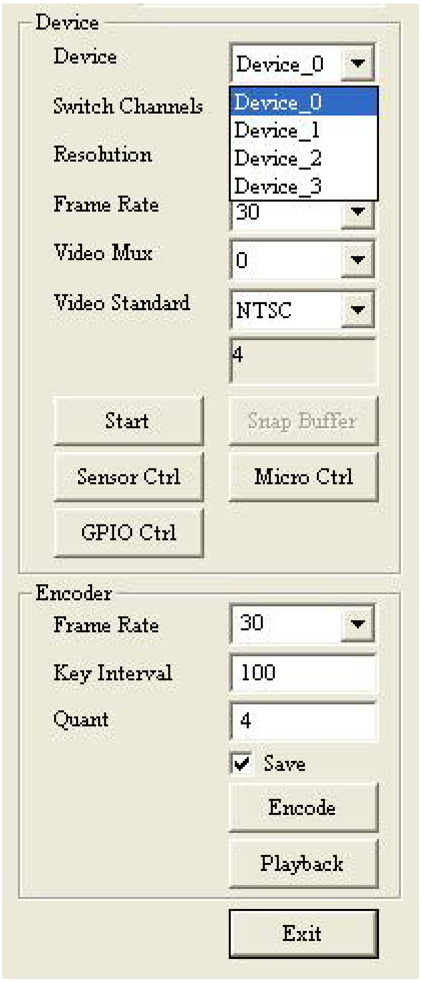
Step 8: There will be a DVMB554.exe icon on the desktop. Execute the demo program.

6.3 Demo Program Functionality

Below is the demo program window. The left side panels are the preview windows of video inputs. The right side panels are the function parameter settings.

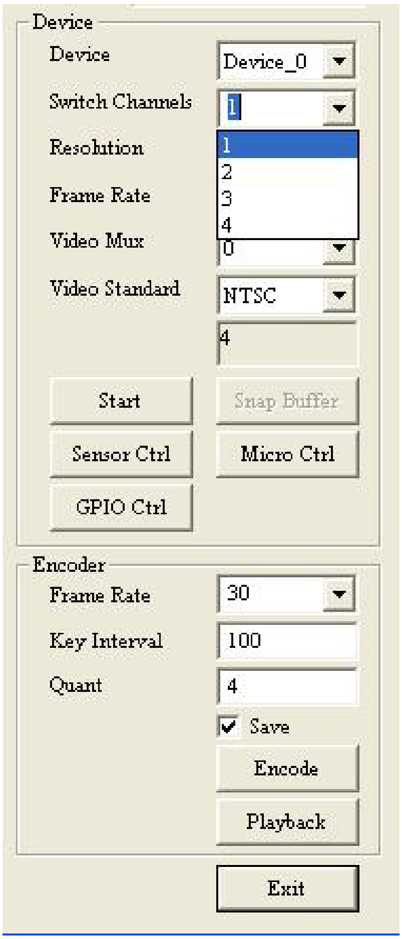
6.3.1 Device

Each device is representative of one Conexant Fusion 878A video capture chip. User can set different parameters to different 878A chip.



6.3.2 Switch Channels

Set the "Switch Channels" to decide how many input for each 878A video chip. Each 878A chip can switch to 4 channel video inputs to share 30/25 frame per second. For more information, please refer to DVMB554SDK_Manual


Device

Device: Device_0

Switch Channels: 1 (1, 2, 3, 4)

Resolution

Frame Rate

Video Mux: 0

Video Standard: NTSC

4

Encoder

Frame Rate: 30

Key Interval: 100

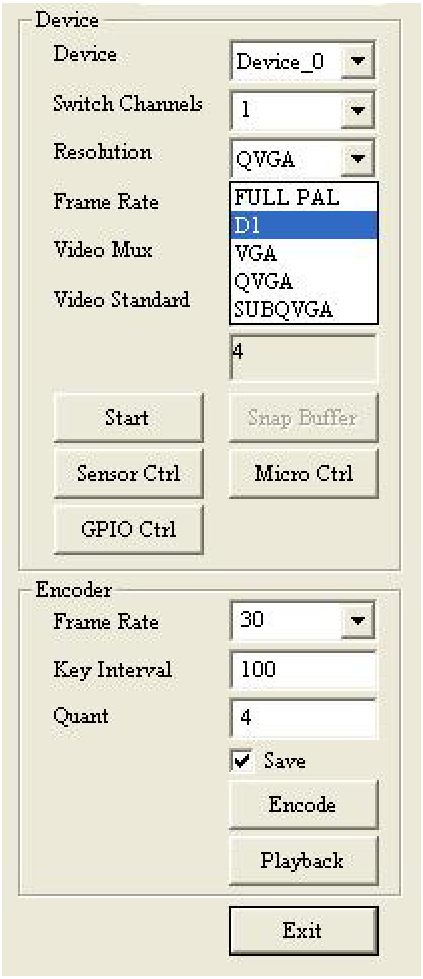
Quant: 4

[✓] Save

6.3.3 Resolution

Set the video capturing resolution. Please refer to DVMB554SDK_Manual

Notice: *For the resolution of VGA or D1, the capture video will have the interlace effect on the video image. In other words, there will be lines in the capture image especially when the targeted image is moving. To eliminate this effect, user might need to set the resolution down to 640x240 and use specific algorisms to compensate the image interlace between the scanning even field image and odd field image. For CIF/320x240 resolution, there will be no interlace effect.*


Device

Device: Device_0

Switch Channels: 1

Resolution: QVGA

FULL PAL
D1
VGA
QVGA
SUBQVGA

Frame Rate

Video Mux

Video Standard

4

Encoder

Frame Rate: 30

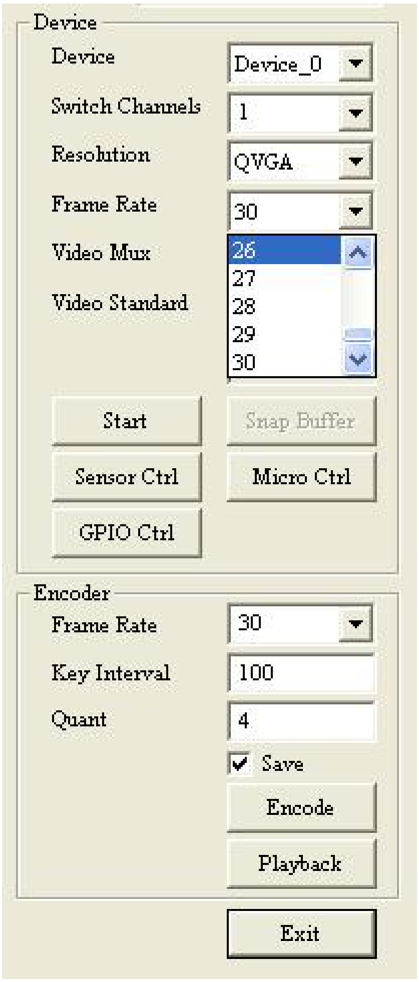
Key Interval: 100

Quant: 4

[✓] Save

6.3.4 Frame Rate

Set the frame rate for video capturing for specific channel. Please refer to DVMB554SDK_Manual


Device

Device: Device_0

Switch Channels: 1

Resolution: QVGA

Frame Rate: 30

Video Mux: 26, 27, 28, 29, 30

Video Standard

Encoder

Frame Rate: 30

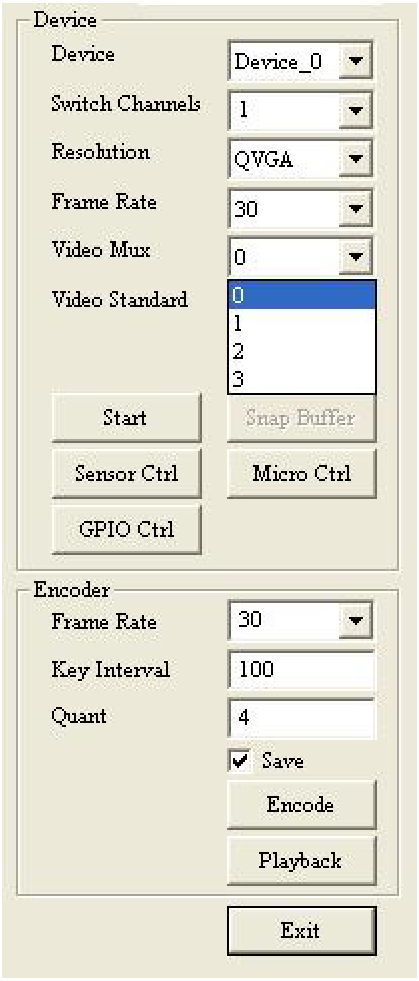
Key Interval: 100

Quant: 4

[✓ Save]

6.3.5 Video Mux

Set the “Video Mux” to specify the video input channel for setting parameter. Please refer to DVMB554SDK_Manual


Device

Device: Device_0

Switch Channels: 1

Resolution: QVGA

Frame Rate: 30

Video Mux: 0

Video Standard: 0

Encoder

Frame Rate: 30

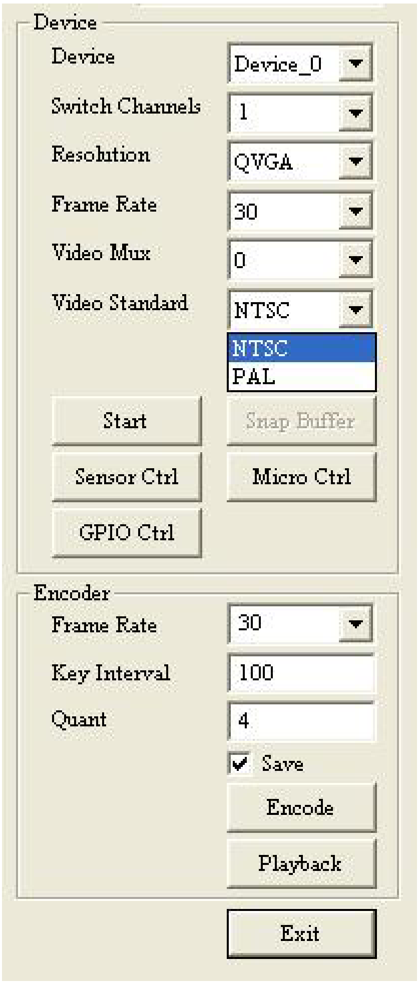
Key Interval: 100

Quant: 4

[✓ Save]

6.3.6 Video Standard

Set the video standard of your cameras. Please refer to DVMB554SDK_Manual.


Device

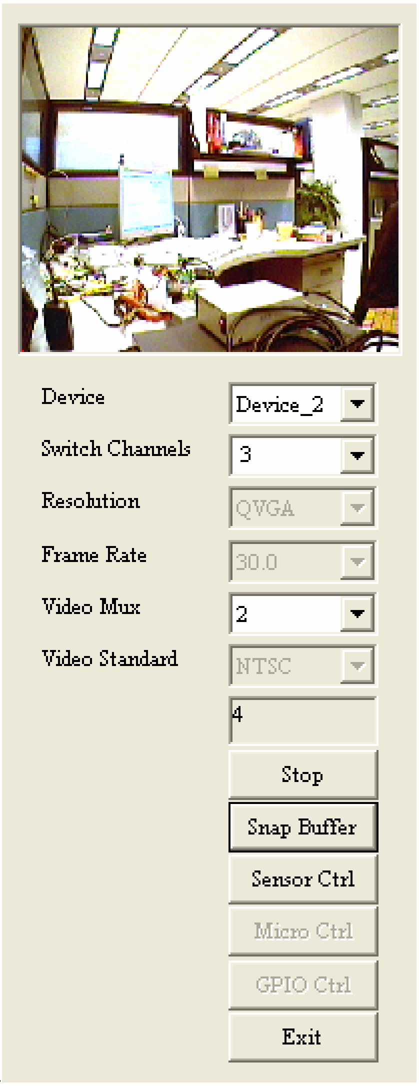
Device: Device_0
Switch Channels: 1
Resolution: QVGA
Frame Rate: 30
Video Mux: 0
Video Standard: NTSC

Encoder

Frame Rate: 30
Key Interval: 100
Quant: 4
[✓ Save]

6.3.7 Snap Buffer

Press the “Snap Buffer” to get the image data of specific channel video input. The snap image will be show on the up panel.


Device: Device_2
Switch Channels: 3
Resolution: QVGA
Frame Rate: 30.0
Video Mux: 2
Video Standard: NTSC
4

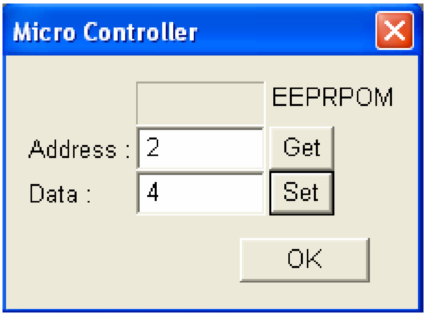
.

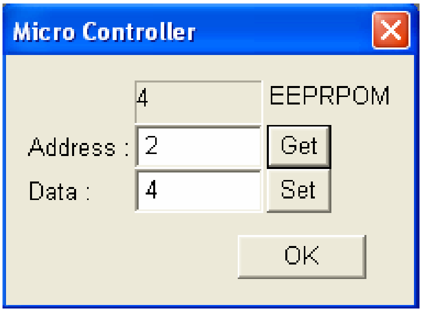
6.3.8 Sensor Control

To set the brightness, contrast, hue and saturation of specific channel.
Please refer to DVMB554SDK_Manual

6.3.9 Micro Control

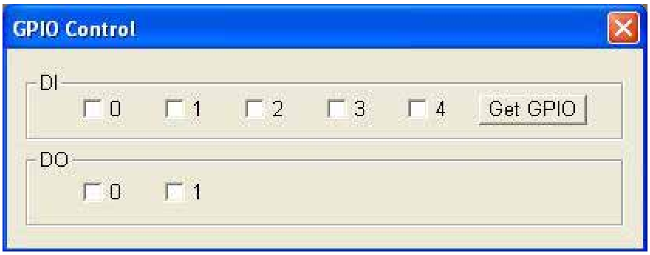
Specify or get the word address(0~127).with a value. Please refer to DVMB554SDK_Manual





6.3.10 GPIO control

To get a specified DI value or to set a specified DO value. Please refer to DVMB554SDK_Manual



A

DVMB-554E SDK Manual

DVMB554 Functions Library

Library: DVMB554.dll

Data Type Summary

Res	The method returned code

Method Summary

SDK Initialize and close	
AdvDVP_CreateSDKInstence	Creates SDK instance
AdvDVP_ReleaseSDKInstence	Releases SDK instance
AdvDVP_InitSDK	Initializes all DVMB554capture devices
AdvDVP_CloseSDK	Cleans all instances of capture devices and closes up the SDK.

Capture control	
AdvDVP_GetNoOfDevices	Gets number of DVP300 Capture Devices
AdvDVP_Start	Starts video capturing
AdvDVP_Stop	Stops video capturing
AdvDVP_GetCapState	Gets capture state
AdvDVP_SetNewFrameCallback	Sets a callback function to SDK
AdvDVP_GetCurFrameBuffer	Gets current frame buffer

Capture setting	
AdvDVP_GetVideoFormat	Gets video input format
AdvDVP_SetVideoFormat	Sets video input format
AdvDVP_GetFrameRate	Gets frame rate
AdvDVP_SetFrameRate	Sets frame rate
AdvDVP_GetResolution	Gets video resolution
AdvDVP_SetResolution	Sets video resolution

Sensor Control	
AdvDVP_GetBrightness	Gets brightness value
AdvDVP_SetBrightness	Sets brightness value
AdvDVP_GetContrast	Gets contrast value
AdvDVP_SetContrast	Sets contrast value
AdvDVP_GetHue	Gets hue value
AdvDVP_SetHue	Sets hue value
AdvDVP_GetSaturation	Gets saturation value
AdvDVP_SetSaturation	Sets saturation value

GPIO	
AdvDVP_InitGPIO	Initializes the GPIO device
AdvDVP_CloseGPIO	Closes the GPIO device
AdvDVP_SetGPIO	Sets value of specified DO pin
AdvDVP_GetGPIO	Gets value of specified DI pin

Micro Controller	
AdvDVP_GetEEData	Reads the value at specified EE word address
AdvDVP_SetEEData	Writes the value at specified EE word address

DVMB554 Encoding Functions Library

Library: DVMB554Encoder.dll

Encoder: rmp4.dll

Before using the DVMB554 encoding functions library, the "RMP4" codec must be installed to the system. After installing the sample program, the codec will be installed automatically. You can install the codec manually by using the "rmp4.inf" file. Right click on the file, and then click "Install".

Data Type Summary

EncRes	The method returned code
PSTREAMREADBEGIN	The stream Read Begin function pointer
PSTREAMREADPROC	The Stream Read Process function pointer
PSTREAMREADEND	The Stream Read End function pointer
STREAMREAD_STRUCT	The structure stores the Stream Read callback function pointers

Method Summary

SDK Initialize and close	
AdvDVP_CreateEncSDKInstence	Creates encoding SDK instance
AdvDVP_ReleaseEncSDKInstence	Releases encoding SDK instance
AdvDVP_InitSDK	Initializes the SDK
AdvDVP_CloseSDK	Closes up the SDK
AdvDVP_InitEncoder	Opens and initializes video encoder
AdvDVP_CloseEncoder	Closes and release video encoder

Encode control	
AdvDVP_StartVideoEncode	Starts video encoding
AdvDVP_VideoEncode	Encodes one video frame
AdvDVP_StopVideoEncode	Stops video encoding
AdvDVP_GetState	Gets encoder state
AdvDVP_CreateAVIFile	Creates an AVI file
AdvDVP_WriteAVIFile	Writes video data to the AVI file
AdvDVP_CloseAVIFile	Closes AVI file
AdvDVP_SetStreamReadCB	Sets the stream read callback functions to SDK

Encode setting	
AdvDVP_GetVideoQuant	Gets video encoding quant
AdvDVP_SetVideoQuant	Sets video encoding quant
AdvDVP_GetVideoFrameRate	Gets video encoding frame rate
AdvDVP_SetVideoFrameRate	Sets video encoding frame rate
AdvDVP_GetVideoResolution	Gets video encoding resolution
AdvDVP_SetVideoResolution	Sets video encoding resolution
AdvDVP_GetVideoKeyInterval	Gets video encoding key interval
AdvDVP_SetVideoKeyInterval	Sets video encoding key interval

DVS 300 Playback Functions Library

Library: DVMB554.dll

Decoder: rmp4.dll

Before using the DVMB554 playback functions library, the "RMP4" codec must be installed to the system. After installing the sample program, the codec will be installed automatically. You can install the codec manually by using the "rmp4.inf" file. Right click on the file, and then click "Install". And, register the decoder filter by using command "regsvr32 dsrmp4.dll".

Data Type Summary

PlayerRes	The method returned code

Method Summary

Playback SDK initialize	
AdvDVP_CreatePlayerSDKInstence	Creates Playback SDK instance
AdvDVP_ReleasePlayerSDKInstence	Releases Playback SDK instance

Playback control	
AdvDVP_OpenFile	Opens file and initialize player
AdvDVP_CloseFile	Closes file that has been opened
AdvDVP_Play	Plays file that has been opened
AdvDVP_Pause	Pauses or continues
AdvDVP_Stop	Stops to play file
AdvDVP_Fast	Plays file with faster speed
AdvDVP_Slow	Plays file with slower speed
AdvDVP_PlayStep	Plays by single frame

AdvDVP_GetStatus	Gets playback state
AdvDVP_GetCurImage	Gets frame that is rendered
AdvDVP_RegNotifyMsg	Registers message sent to player when event occurs
AdvDVP_CheckFileEnd	Checks if file is finished playing

Playback setting	
AdvDVP_GetVideoResolution	Gets video resolution of file
AdvDVP_GetFileTime	Gets total file time
AdvDVP_GetPlayedTime	Gets current file time
AdvDVP_SetPlayPosition	Locates position of file
AdvDVP_GetFileTotalFrames	Gets total frame number of file
AdvDVP_GetPlayedFrames	Gets current frame number of file
AdvDVP_GetPlayRate	Gets current played rate

DVMB554 Functions Reference

Data Type

Res

Syntax

```
typedef enum tagRes
{
    SUCCEEDED           = 1,
    FAILED              = 0,
    SDKINITFAILED       = -1,
    PARAMERROR          = -2,
    NODEVICES           = -3,
    NOSAMPLE            = -4,
    DEVICENUMERROR      = -5,
    INPUTERROR          = -6,
    VERIFYHWERROR       = -7
} Res;
```

Description

The method returned code.

Method

AdvDVP_CreateSDKInstence

Syntax

int AdvDVP_CreateSDKInstence(void **pp)

Parameters

pp:	A pointer to the SDK.

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
PARAMERROR:	Parameter error.

Description

This function creates SDK instance.

See Also

AdvDVP_ReleaseSDKInstence

AdvDVP_ReleaseSDKInstence

Syntax

int AdvDVP_ReleaseSDKInstence(void *p)

Parameters

p:	The SDK instance is created by “AdvDVP_CreateSDKInstence” function.

Return Value

SUCCEEDED:	Function succeeded.

Description

This function releases SDK instance created by the “AdvDVP_CreateSDKInstence” function.

See Also

AdvDVP_CreateSDKInstence

AdvDVP_InitSDK

Syntax

int AdvDVP_InitSDK()

Parameters

None

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
NODEVICES:	No devices found.
VERIFYHWERROR	Verify the hardware error.

Description

This function initializes all DVMB554 capture devices in the system. After initializing each device, the capture status would be set as “STOPPED”.

See Also

AdvDVP_GetNoOfDevices
AdvDVP_GetCapState
AdvDVP_CloseSDK

AdvDVP_CloseSDK

Syntax

int AdvDVP_CloseSDK(void)

Parameters

None

Return Value

SUCCEEDED:	Function succeeded.
SDKINITFAILED:	SDK not initialized.

Description

This function cleans all instances of capture devices and closes up the SDK.

See Also

AdvDVP_InitSDK

AdvDVP_GetNumberOfDevices

Syntax

int AdvDVP_GetNoOfDevices(int *pNoOfDevs)

Parameters

pNoOfDevs:	A pointer to get number of DVMB554 Capture Devices.

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
SDKINITFAILED:	SDK not initialized.

Description

This function gets number of DVMB554 Capture Devices in the system. At most 16 channels are available in a DVMB554 system.

AdvDVP_Start

Syntax

int AdvDVP_Start(int nDevNum, HWND Main, HWND hwndPreview)

Parameters

nDevNum:	Specifies the device number(0~3).
Main:	A main window handle.
hwndPreview:	A windows handle for display area. When the value of this parameter is NULL, the video will not be rendered.

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
DEVICENUMERROR:	Invalid device number.
SDKINITFAILED:	SDK not initialized.

Description

This function starts video capturing on a specified capture port. The capture state would be set as “RUNNING” after a successful start.

See Also

AdvDVP_Stop
AdvDVP_GetCapState

AdvDVP_Stop

Syntax

int AdvDVP_Stop(int nDevNum)

Parameters

nDevNum:	Specifies the device number(0~3).

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
DEVICENUMERROR:	Invalid device number.
SDKINITFAILED:	SDK not initialized.

Description

This function stops video capturing on a specified capture port. The capture state would be set as “STOPPED” after a successful stop.

See Also

AdvDVP_Start
AdvDVP_GetCapState

AdvDVP_GetCapState

Syntax

int AdvDVP_GetCapState(int nDevNum)

Parameters

nDevNum:	Specifies the device number(0~3).

Return Value

DEVICENUMERROR:	Invalid device number.
SDKINITFAILED:	SDK not initialized.

Description

This function gets capture state of a specified capture port.

```
typedef enum {
    STOPPED          = 1,
    RUNNING          = 2,
    UNINITIALIZED    = -1,
    UNKNOWNSTATE     = -2
} CapState;
```

See Also

AdvDVP_InitSDK
AdvDVP_Start
AdvDVP_Stop

AdvDVP_GetCurFrameBuffer

Syntax

int AdvDVP_GetCurFrameBuffer(int nDevNum, long* bufSize, BYTE* buf)

Parameters

nDevNum:	Specifies the device number(0~3).
bufSize:	Frame buffer size.
buf:	Frame buffer.

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
DEVICENUMERROR:	Invalid device number.
PARAMERROR:	Invalid parameter.
SDKINITFAILED:	SDK not initialized.
NOSAMPLE:	No buffer sample.

Description

This function gets current frame buffer of a specified capture port. Start capturing before the function is called.

See Also

AdvDVP_Start

AdvDVP_SetNewFrameCallback

Syntax

int AdvDVP_SetNewFrameCallback(int nDevNum, int callback)

Parameters

nDevNum:	Specifies the device number(0~3).
callback:	Callback function.

Callback fumction type:

typedef int (*CAPCALLBACK)(int nDevNum, int bufsize, BYTE* buf);

nDevNum:	Specifies the device number(0~3).
bufsize:	An integer pointer of the frame buffer size.
buf:	A BYTE pointer of the frame buffer.

Return Value

SUCCEEDED:	Function succeeded.
DEVICENUMERROR:	Invalid device number.
SDKINITFAILED:	SDK not initialized.

Description

This function sets a callback function to SDK. When new frame arrived, messages and frame information will be sent to callback function.

See Also

AdvDVP_GetVideoFormat

Syntax

int AdvDVP_GetVideoFormat(int nDevNum, AnalogVideoFormat* vFormat)

Parameters

nDevNum:	Specifies the device number(0~3).
Vformat:	A pointer to get video format.

```
typedef enum tagAnalogVideoFormat
{
    Video_None          = 0x00000000,
    Video_NTSC_M        = 0x00000001,
    Video_NTSC_M_J      = 0x00000002,
    Video_PAL_B         = 0x00000010,
    Video_PAL_M         = 0x00000200,
    Video_PAL_N         = 0x00000400,
    Video_SECAM_B       = 0x00001000
} AnalogVideoFormat;
```

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
DEVICENUMERROR:	Invalid device number.
PARAMERROR:	Invalid parameter.
SDKINITFAILED:	SDK not initialized.

Description

This function gets video input format of a specified capture port.

See Also

AdvDVP_SetVideoFormat

AdvDVP_SetVideoFormat

Syntax

int AdvDVP_SetVideoFormat(int nDevNum, AnalogVideoFormat* vFormat)

Parameters

nDevNum:	Specifies the port device number(0~3).
Vformat:	video format:

```
typedef enum tagAnalogVideoFormat
{
    Video_None          = 0x00000000,
    Video_NTSC_M        = 0x00000001,
    Video_NTSC_M_J      = 0x00000002,
    Video_PAL_B         = 0x00000010,
    Video_PAL_M         = 0x00000200,
    Video_PAL_N         = 0x00000400,
    Video_SECAM_B       = 0x00001000
} AnalogVideoFormat;
```

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
DEVICENUMERROR:	Invalid device number.
SDKINITFAILED:	SDK not initialized.

Description

This function sets video input format a specified capture port. This function should be called before "AdvDVP_Start".

See Also

AdvDVP_GetVideoFormat

AdvDVP_GetFrameRate

Syntax
int AdvDVP_GetFrameRate(int nDevNum, int *FrameRate)

Parameters

nDevNum:	Specifies the device number(0~3).
FrameRate:	A pointer to get video frame rate.

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
DEVICENUMERROR:	Invalid device number.
PARAMERROR:	Invalid parameter.
SDKINITFAILED:	SDK not initialized.

Description
This function gets frame rate of a specified capture port.

See Also
AdvDVP_SetFrameRate

AdvDVP_SetFrameRate

Syntax

```
int AdvDVP_SetFrameRate(int nDevNum, int  FrameRate)
```

Parameters

nDevNum:	Specifies the device number(0~3).
FrameRate:	A value to set frame rate. (0<FrameRate<=30, Default value is 30)

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
DEVICENUMERROR:	Invalid device number.
PARAMERROR:	Invalid parameter.
SDKINITFAILED:	SDK not initialized.

Description

This function sets frame rate of a specified capture port. This function should be called before “AdvDVP_Start”.

See Also

AdvDVP_GetFrameRate

AdvDVP_GetResolution

Syntax

int AdvDVP_GetResolution(int nDevNum, VideoSize *Size)

Parameters

nDevNum:	Specifies the device number(0~3).
Size:	A pointer to get video resolution.

```
typedef enum
{
    SIZED1=0,          // (NTSC: 720x480, PAL: 720x576)
    SIZEVGA,           //(640x480)
    SIZEQVGA,          //(320x240)
    SIZESUBQVGA        //(160x120)
} VideoSize;
```

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
DEVICENUMERROR:	Invalid device number.
PARAMERROR:	Invalid parameter.
SDKINITFAILED:	SDK not initialized.

Description

This function gets video resolution of a specified capture port.

See Also

AdvDVP_SetResolution

AdvDVP_SetResolution

Syntax

```
int AdvDVP_SetResolution(int nDevNum, VideoSize Size)
```

Parameters

nDevNum:	Specifies the device number(0~3).
Size:	A value to set video resolution.

```
typedef enum
{
    SIZED1=0,          // (NTSC: 720x480, PAL: 720x576)
    SIZEVGA,           //(640x480)
    SIZEQVGA,          //(320x240)
    SIZESUBQVGA        //(160x120)
} VideoSize;
```

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
DEVICENUMERROR:	Invalid device number.
SDKINITFAILED:	SDK not initialized.

Description

This function sets video resolution of a specified capture port. This function should be called before “AdvDVP_Start”.

See Also

AdvDVP_GetResolution

AdvDVP_GetBrightness

Syntax

AdvDVP_GetBrightness(int nDevNum, long *lpValue)

Parameters

nDevNum:	Specifies the device number(0~3).
lpValue:	A long pointer to get brightness value.

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
DEVICENUMERROR:	Invalid device number.
PARAMERROR:	Invalid parameter.
SDKINITFAILED:	SDK not initialized.

Description

This function gets brightness value of a specified capture port.

See Also

AdvDVP_SetBrightness

AdvDVP_SetBrightness

Syntax

int AdvDVP_SetBrightness(int nDevNum, long lValue)

Parameters

nDevNum:	Specifies the device number(0~3).
lValue:	A value to set brightness(0~100).

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
DEVICENUMERROR:	Invalid device number.
PARAMERROR:	Invalid parameter.
SDKINITFAILED:	SDK not initialized.

Description

This function sets brightness value of a specified capture port.

See Also

AdvDVP_GetBrightness

AdvDVP_GetContrast

Syntax

int AdvDVP_GetContrast(int nDevNum, long *lpValue)

Parameters

nDevNum:	Specifies the device number(0~3).
lpValue:	A long pointer to get contrast value.

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
DEVICENUMERROR:	Invalid device number.
PARAMERROR:	Invalid parameter.
SDKINITFAILED:	SDK not initialized.

Description

This function gets contrast value of a specified capture port.

See Also

AdvDVP_SetContrast

AdvDVP_SetContrast

Syntax

int AdvDVP_SetContrast(int nDevNum, long lValue)

Parameters

nDevNum:	Specifies the device number(0~3).
lValue:	A value to set contrast(0~100).

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
DEVICENUMERROR:	Invalid device.
PARAMERROR:	Invalid parameter.
SDKINITFAILED:	SDK not initialized.

Description

This function sets contrast value of a specified capture port.

See Also

AdvDVP_GetContrast

AdvDVP_GetHue

Syntax

int AdvDVP_GetHue(int nDevNum, long *lpValue)

Parameters

nDevNum:	Specifies the device number(0~3).
lpValue:	A long pointer to get hue value.

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
DEVICENUMERROR:	Invalid device number.
PARAMERROR:	Invalid parameter.
SDKINITFAILED:	SDK not initialized.

Description

This function gets hue value of a specified capture port.

See Also

AdvDVP_SetHue

AdvDVP_SetHue

Syntax

int AdvDVP_SetHue(int nDevNum, long lValue)

Parameters

nDevNum:	Specifies the device number(0~3).
lValue:	A value to set hue(0~100).

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
DEVICENUMERROR:	Invalid device number.
PARAMERROR:	Invalid parameter.
SDKINITFAILED:	SDK not initialized.

Description

This function sets hue value of a specified capture port.

See Also

AdvDVP_GetHue

AdvDVP_GetSaturation

Syntax

```
int AdvDVP_GetSaturation(int nDevNum, long *lpValue)
```

Parameters

nDevNum:	Specifies the device number(0~3).
lpValue:	A long pointer to get saturation value.

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
DEVICENUMERROR:	Invalid device number.
PARAMERROR:	Invalid parameter.
SDKINITFAILED:	SDK not initialized.

Description

This function gets saturation value of a specified capture port.

See Also

AdvDVP_SetSaturation

AdvDVP_SetSaturation

Syntax

int AdvDVP_SetSaturation(int nDevNum, long lValue)

Parameters

nDevNum:	Specifies the device number(0~3).
lValue:	A value to set saturation(0~100).

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
DEVICENUMERROR:	Invalid device number.
PARAMERROR:	Invalid parameter.
SDKINITFAILED:	SDK not initialized.

Description

This function sets saturation value of a specified capture port.

See Also

AdvDVP_GetSaturation

AdvDVP_InitGPIO

Syntax

int AdvDVP_InitGPIO()

Parameters

None.

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
SDKINITFAILED:	SDK not initialized.

Description

This function initializes the GPIO device. The function must be called to initialize the GPIO device before using other GPIO functions.

See Also

AdvDVP_CloseGPIO
AdvDVP_SetGPIO
AdvDVP_GetGPIO

AdvDVP_CloseGPIO

Syntax
```
int AdvDVP_CloseGPIO()
```

Parameters
None.

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
SDKINITFAILED:	SDK not initialized.

Description
This function closes the GPIO device. The function must be called to release GPIO device after finishing all GPIO functions.

See Also
AdvDVP_InitGPIO
AdvDVP_SetGPIO
AdvDVP_GetGPIO

AdvDVP_SetGPIO

Syntax

int AdvDVP_SetGPIO(int nDONum, BOOL bValue)

Parameters

nDONum:	Specifies the digital output number(0~7).
bValue:	A value to set the value of the specified digital output.

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
PARAMERROR:	Invalid parameter.
SDKINITFAILED:	SDK not initialized.

Description

This function sets the value of the specified digital output.

See Also

AdvDVP_InitGPIO
AdvDVP_CloseGPIO
AdvDVP_GetGPIO

AdvDVP_GetGPIO

Syntax

int AdvDVP_GetGPIO(int nDINum, BOOL *pbValue)

Parameters

nDINum:	Specifies the digital input number(0~5).
pbValue:	A pointer to get the value of the specified digital input.

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
PARAMERROR:	Invalid parameter.
SDKINITFAILED:	SDK not initialized.

Description

This function gets the value of the specified digital input.

See Also

AdvDVP_InitGPIO
AdvDVP_CloseGPIO
AdvDVP_SetGPIO

AdvDVP_GetEEData

Syntax

```
int AdvDVP_GetEEData(BYTE wordAddr, BYTE* pData)
```

Parameters

wordAddr:	Specifies the word address(0~127).
pData:	A pointer to get byte value stored in EE.

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
PARAMERROR:	Invalid parameter.
SDKINITFAILED:	SDK not initialized.

Description

This function read the value at specified EE word address.

See Also

AdvDVP_SetEEData

AdvDVP_SetEEData

Syntax

int AdvDVP_SetEEData(BYTE wordAddr, BYTE* pData)

Parameters

wordAddr:	Specifies the word address(0~127).
pData:	A value to set the byte value in EE.

Return Value

SUCCEEDED:	Function succeeded.
FAILED:	Function failed.
PARAMERROR:	Invalid parameter.
SDKINITFAILED:	SDK not initialized.

Description

This function writes the value at specified EE word address.

See Also

AdvDVP_GetEEData

DVMB554 Encoding Functions Reference

Data Type

EncRes

Syntax

```
typedef enum tagRes
{
     ENC_SUCCEEDED        = 1,
     ENC_FAILED           = 0,
     ENC_SDKINITFAILED    = -1,
     ENC_ENCINITFAILED    = -2,
     ENC_PARAMERROR       = -3,
     ENC_VERIFYHWERROR    = -4,
     ENC_ENCNUMERROR      = -5,
     ENC_BUFFERFULL       = -6
} EncRes;
```

Description

The method returned code.

PSTREAMREADBEGIN

Syntax

void (*PSTREAMREADBEGIN)(int nEncNum)

Parameters

nEncNum:	Specifies the encoder number.

Return Value

None

Description

The pointer to the Stream Read Begin callback function called when begins the video stream read process.

See Also

STREAMREAD_STRUCT

PSTREAMREADPROC

Syntax

void (*PSTREAMREADPROC)(int nEncNum, LPVOID pStreamBuf, long lBufSize, DWORD dwCompFlags)

Parameters

nEncNum:	Specifies the encoder number.
pStreamBuf:	A point to the data buffer stores an encoded video frame.
lBufSize:	Specifies the size of the encoded video frame.
dwCompFlags	Specifies if this encoded video frame is I-frame. The AVIIF_KEYFRAME value means the frame is I-frame.

```
#define AVIIF_KEYFRAME   0x00000010L
```

Return Value

None

Description

The pointer to the Stream Read Process callback function called after every video frame is encoded. User can use this function to get every encoded video frame.

See Also

STREAMREAD_STRUCT

PSTREAMREADEND

Syntax

void (*PSTREAMREADEND)(int nEncNum)

Parameters

nEncNum:	Specifies the encoder number.

Return Value

None

Description

The pointer to the Stream Read End callback function called when the video stream read process is finished.

See Also

STREAMREAD_STRUCT

STREAMREAD_STRUCT structure

Syntax

```
typedef struct
{
    void (*PSTREAMREADBEGIN)(int nEncNum);
    void (*PSTREAMREADPROC)(int nEncNum, LPVOID
    pStreamBuf, long lBufSize, DWORD dwCompFlags);
    void (*PSTREAMREADEND)(int nEncNum);
}STREAMREAD_STRUCT;
```

Parameters:

PSTREAMREADBEGIN:	The pointer to the Stream Read Begin callback function called when begins the video stream read process.
PSTREAMREADPROC:	The pointer to the Stream Read Process callback function called after every video frame is encoded.
PSTREAMREADEND:	The pointer to the Stream Read End callback function called when the video stream read process is finished.

Description

This structure stores the Stream Read callback function pointers.

See Also

PSTREAMREADBEGIN
PSTREAMREADPROC
PSTREAMREADEND
AdvDVP_SetStreamReadCB

Method

AdvDVP_CreateEncSDKInstence

Syntax

```
int AdvDVP_CreateEncSDKInstence (void **pp)
```

Parameters

pp:	A pointer to the encoding SDK.

Return Value

ENC_SUCCEEDED:	Function succeeded.
ENC_FAILED:	Function failed.
ENC_PARAMERROR:	Parameter error.

Description

This function creates the encoding SDK instance.

See Also

AdvDVP_ReleaseEncSDKInstence

AdvDVP_ReleaseEncSDKInstence

Syntax

int AdvDVP_ReleaseEncSDKInstence(void *p)

Parameters

p:	The encoding SDK instance is created by “AdvDVP_CreateEncSDKInstence” function.

Return Value

SUCCEEDED:	Function succeeded.

Description

This function releases encoding SDK instance created by the “AdvDVP_CreateEncSDKInstence” function.

See Also

AdvDVP_CreateEncSDKInstence

AdvDVP_InitSDK

Syntax
int AdvDVP_InitSDK(void)

Parameters
None

Return Value

ENC_SUCCEEDED:	Function succeeded.
ENC_VERIFYHWERROR	Verify the hardware error.

Description
This function initializes all parameters of the SDK in the system.

See Also
AdvDVP_CloseSDK

AdvDVP_CloseSDK

Syntax

int AdvDVP_CloseSDK(void)

Parameters

None

Return Value

ENC_SUCCEEDED:	Function succeeded.
ENC_SDKINITFAILED:	SDK does not be initialized successfully.

Description

This function cleans all parameters of the SDK and closes up the SDK.

See Also

AdvDVP_InitSDK

AdvDVP_InitEncoder

Syntax

```
int AdvDVP_InitEncoder(int nEncNum, int nEncBufSize)
```

Parameters

nEncNum:	Specifies the encoder number (0~15).
nEncBufSize:	Specifies the encoding buffer size.

Return Value

ENC_SUCCEEDED:	Function succeeded.
ENC_FAILED:	Function failed.
ENC_SDKINITFAILED:	SDK does not be initialized successfully.
ENC_ENCNUMERROR:	Invalid encoder number.

Description

This function opens and initializes the specified video encoder. After initializing the encoder, the encoding state would be set as “ENC_STOPPED”.

See Also

AdvDVP_CloseEncoder
AdvDVP_GetState

AdvDVP_CloseEncoder

Syntax

int AdvDVP_CloseEncoder(int nEncNum)

Parameters

nEncNum:	Specifies the encoder number (0~15).

Return Value

ENC_SUCCEEDED:	Function succeeded.
ENC_FAILED:	Function failed.
ENC_SDKINITFAILED:	SDK does not be initialized successfully.
ENC_ENCNUMERROR:	Invalid encoder number.
ENC_ENCINITFAILED:	Encoder does not be initialized successfully.

Description

This function closes and releases the specified video encoder. After successfully calling this function, the encoding state would be set as “ENC_UNINITIALIZED”.

See Also

AdvDVP_InitEncoder
AdvDVP_GetState

AdvDVP_StartVideoEncode

Syntax

```
int AdvDVP_StartVideoEncode(int nEncNum)
```

Parameters

nEncNum:	Specifies the encoder number (0~15).

Return Value

ENC_SUCCEEDED:	Function succeeded.
ENC_FAILED:	Function failed.
ENC_SDKINITFAILED:	SDK does not be initialized successfully.
ENC_ENCNUMERROR:	Invalid encoder number.
ENC_ENCINITFAILED:	Encoder does not be initialized successfully.

Description

This function notifies the specified video encoder to prepare to encode the video. The encode state would be set as “ENC_RUNNING” after a successful beginning.

See Also

AdvDVP_VideoEncode
AdvDVP_StopVideoEncode
AdvDVP_GetState

AdvDVP_VideoEncode

Syntax

int AdvDVP_VideoEncode(int nEncNum, LPVOID lpInBuf,
int InBufSize, BOOL bKeyFrame)

Parameters

nEncNum:	Specifies the encoder number (0~15).
lpbiIn:	Specifies the input buffer stores the source video frame.
InBufSize:	Specifies the size of the input buffer.
bKeyFrame:	Specifies if the video frame is encoded as a I-frame.

Return Value

ENC_SUCCEEDED:	Function succeeded.
ENC_FAILED:	Function failed.
ENC_SDKINITFAILED:	SDK does not be initialized successfully.
ENC_ENCNUMERROR:	Invalid encoder number.
ENC_ENCINITFAILED:	Encoder does not be initialized successfully.
ENC_PARAMERROR:	Parameter error.
ENC_BUFFERFULL:	Encoding buffer is full, the video frame can not be written to the buffer.

Description

This function writes the video frame to the encoding buffer to encode it by the specified encoder.

See Also

AdvDVP_StartVideoEncode
AdvDVP_StopVideoEncode

AdvDVP_StopVideoEncode

Syntax

int AdvDVP_StopVideoEncode(int nEncNum)

Parameters

nEncNum:	Specifies the encoder number (0~15).

Return Value

ENC_SUCCEEDED:	Function succeeded.
ENC_FAILED:	Function failed.
ENC_SDKINITFAILED:	SDK does not be initialized successfully.
ENC_ENCNUMERROR:	Invalid encoder number.
ENC_ENCINITFAILED:	Encoder does not be initialized successfully.

Description

This function notifies the specified video encoder to stop encoding and releases all relational resources. The encoding state would be set as “ENC_STOPPED” after a successful stop.

See Also

AdvDVP_StartVideoEncode
AdvDVP_VideoEncode
AdvDVP_GetState

AdvDVP_GetState

Syntax

int AdvDVP_GetState(int nEncNum)

Parameters

nEncNum:	Specifies the encoder number (0~15).

Return Value

ENC_ENCNUMERROR:	Invalid encoder number.

Description

This function gets encoding state of a specified video encoder.

```
typedef enum
{
    ENC_STOPPED            = 1,
    ENC_RUNNING            = 2,
    ENC_UNINITIALIZED      = -1,
} EncoderState;
```

See Also

AdvDVP_InitEncoder
AdvDVP_CloseEncoder
AdvDVP_StartVideoEncode
AdvDVP_StopVideoEncode

AdvDVP_SetStreamReadCB

Syntax

void AdvDVP_SetStreamReadCB(STREAMREAD_STRUCT *pStreamRead)

Parameters

pStreamRead:	A pointer to STREAMREAD_STRUCT structure recording the pointers to the StreamRead callback functions.

Return Value

None

Description

This function registers the Stream Read callback functions to the SDK.

See Also

STREAMREAD_STRUCT structure

AdvDVP_GetVideoQuant

Syntax

```
int AdvDVP_GetVideoQuant(int nEncNum, int *nQuant)
```

Parameters

nEncNum:	Specifies the encoder number (0~15).
nQuant:	A pointer to get the video quant. The range is 1~31. The default video quality is 4.

Return Value

ENC_SUCCEEDED:	Function succeeded.
ENC_FAILED:	Function failed.
ENC_SDKINITFAILED:	SDK does not be initialized successfully.
ENC_ENCNUMERROR:	Invalid encoder number.
ENC_ENCINITFAILED:	Encoder does not be initialized successfully.

Description

This function gets video quant of the specified video encoder. The lower video quant can get the compressed video with higher quality and bit rate, vice versa.

See Also

AdvDVP_SetVideoQuant

AdvDVP_SetVideoQuant

Syntax

int AdvDVP_SetVideoQuant(int nEncNum, int nQuant)

Parameters

nEncNum:	Specifies the encoder number (0~15).
nQuant:	A value to set the video quant. The range is 1~31. The default video quality is 4.

Return Value

ENC_SUCCEEDED:	Function succeeded.
ENC_FAILED:	Function failed.
ENC_SDKINITFAILED:	SDK does not be initialized successfully.
ENC_ENCNUMERROR:	Invalid encoder number.
ENC_ENCINITFAILED:	Encoder does not be initialized successfully.

Description

This function sets video quant of the specified video encoder. The lower video quant can get the compressed video with higher quality and bit rate, vice versa.

See Also

AdvDVP_GetVideoQuant

AdvDVP_GetVideoFrameRate

Syntax

```
int AdvDVP_GetVideoFrameRate(int nEncNum, int *nFrameRate)
```

Parameters

nEncNum:	Specifies the encoder number (0~15).
nFrameRate:	A pointer to get the video frame rate.

Return Value

ENC_SUCCEEDED:	Function succeeded.
ENC_FAILED:	Function failed.
ENC_SDKINITFAILED:	SDK does not be initialized successfully.
ENC_ENCNUMERROR:	Invalid encoder number.
ENC_ENCINITFAILED:	Encoder does not be initialized successfully.

Description

This function gets video frame rate of the specified video encoder.

See Also

AdvDVP_SetVideoFrameRate

AdvDVP_SetVideoFrameRate

Syntax

int AdvDVP_SetVideoFrameRate(int nEncNum, int nFrameRate)

Parameters

nEncNum:	Specifies the encoder number (0~15).
nFrameRate:	A value to set the video frame rate. The range is 1~30. The default video frame rate is 30.

Return Value

ENC_SUCCEEDED:	Function succeeded.
ENC_FAILED:	Function failed.
ENC_SDKINITFAILED:	SDK does not be initialized successfully.
ENC_ENCNUMERROR:	Invalid encoder number.
ENC_ENCINITFAILED:	Encoder does not be initialized successfully.

Description

This function sets video frame rate of the specified video encoder.

See Also

AdvDVP_GetVideoFrameRate

AdvDVP_GetVideoResolution

Syntax

int AdvDVP_GetVideoResolution(int nEncNum, int *nWidth, int *nHeight)

Parameters

nEncNum:	Specifies the encoder number (0~15).
nWidth:	A pointer to get the width of the video.
nHeight:	A pointer to get the height of the video.

Return Value

ENC_SUCCEEDED:	Function succeeded.
ENC_SDKINITFAILED:	SDK does not be initialized successfully.
ENC_ENCNUMERROR:	Invalid encoder number.
ENC_ENCINITFAILED:	Encoder does not be initialized successfully.

Description

This function gets video resolution of the specified video encoder.

See Also

AdvDVP_SetVideoResolution

AdvDVP_SetVideoResolution

Syntax

int AdvDVP_SetVideoResolution(int nEncNum, int nWidth, int nHeight)

Parameters

nEncNum:	Specifies the encoder number (0~15).
nWidth:	A value to set the width of the video. The default width is 320.
nHeight	A value to set the height of the video. The default height is 240.

Return Value

ENC_SUCCEEDED:	Function succeeded.
ENC_FAILED:	Function failed.
ENC_SDKINITFAILED:	SDK does not be initialized successfully.
ENC_ENCNUMERROR:	Invalid encoder number.
ENC_ENCINITFAILED:	Encoder does not be initialized successfully.

Description

This function sets video resolution of the specified video encoder.

See Also

AdvDVP_GetVideoResolution

AdvDVP_GetVideoKeyInterval

Syntax

```
int AdvDVP_GetVideoKeyInterval(int nEncNum,
int *nKeyInterval)
```

Parameters

nEncNum:	Specifies the encoder number (0~15).
nKeyInterval:	A pointer to get the interval of the video key frame.

Return Value

ENC_SUCCEEDED:	Function succeeded.
ENC_FAILED:	Function failed.
ENC_SDKINITFAILED:	SDK does not be initialized successfully.
ENC_ENCNUMERROR:	Invalid encoder number.
ENC_ENCINITFAILED:	Encoder does not be initialized successfully.

Description

This function gets the interval of the video key frame of the specified video encoder.

See Also

AdvDVP_SetVideoKeyInterval

AdvDVP_SetVideoKeyInterval

Syntax
int AdvDVP_SetVideoKeyInterval(int nEncNum, int nKeyInterval)

Parameters

nEncNum:	Specifies the encoder number (0~15).
nKeyInterval:	A value to set the interval of the video key frame. The range is 1~99. The default video frame rate is 60.

Return Value

ENC_SUCCEEDED:	Function succeeded.
ENC_FAILED:	Function failed.
ENC_SDKINITFAILED:	SDK does not be initialized successfully.
ENC_ENCNUMERROR:	Invalid encoder number.
ENC_ENCINITFAILED:	Encoder does not be initialized successfully.

Description
This function sets the interval of the video key frame of the specified video encoder.

See Also
AdvDVP_GetVideoKeyInterval

AdvDVP_CreateAVIFile

Syntax

HANDLE AdvDVP_CreateAVIFile(LPCSTR lpcsFileName, int nWidth, int nHeight, int nFrameRate)

Parameters

lpcsFileName:	Specifies the file name of the AVI file.
nWidth:	
nHeight	
nFrameRate	Specifies the frame rate of the video.

Return Value

If the function succeeds, the file handle is returned. Otherwise, the function returns NULL.

Description

This function creates the AVI file to save the encoded video stream.

See Also

AdvDVP_WriteAVIFile
AdvDVP_CloseAVIFile

AdvDVP_WriteAVIFile

Syntax

int AdvDVP_WriteAVIFile(HANDLE hAVIFile, LPVOID lpStreamBuf, long lBufSize, DWORD dwCompFlags)

Parameters

hAVIFile:	Specifies the AVI file handle.
lpStreamBuf:	A pointer to the video stream data buffer written into the file.
lBufSize:	Specifies the size of the video stream data buffer.
dwCompFlags:	Flag associated with this data. The AVIIF_KEYFRAME flag is defined to indicate this data does not rely on preceding data in the file.

#define AVIIF_KEYFRAME   0x00000010L

Return Value

ENC_SUCCEEDED:	Function succeeded.
ENC_FAILED:	Function failed.
ENC_SDKINITFAILED:	SDK does not be initialized successfully.

Description

This function writes the video stream data into the specified AVI file.

See Also

AdvDVP_CreateAVIFile
AdvDVP_CloseAVIFile

AdvDVP_CloseAVIFile

Syntax

```
int AdvDVP_CloseAVIFile(HANDLE hAVIFile)
```

Parameters

hAVIFile:	Specifies the AVI file handle.

Return Value

ENC_SUCCEEDED:	Function succeeded.
ENC_FAILED:	Function failed.
ENC_SDKINITFAILED:	SDK does not be initialized successfully.

Description

This function closes the specified AVI file.

See Also

AdvDVP_CreateAVIFile
AdvDVP_WriteAVIFile

Playback Functions Reference

Data Type

PlayerRes

Syntax

```
typedef enum tagRes
{
    PLAYER_SUCCEEDED            = 1,
    PLAYER_FAILED               = 0,
    PLAYER_SDKINITFAILED        = -1,
    PLAYER_PARAMERROR           = -2,
} PlayerRes;
```

Description

The method returned code.

Method

AdvDVP_CreatePlayerSDKInstence

Syntax

int AdvDVP_CreatePlayerSDKInstence(void **pp)

Parameters

pp:	A pointer to the playback SDK.

Return Value

PLAYER_SUCCEEDED:	Function succeeded.
PLAYER_FAILED:	Function failed.
PLAYER_PARAMERROR:	Parameter error.

Description

This function creates playback SDK instance.

See Also

AdvDVP_ReleasePlayerSDKInstence

AdvDVP_ReleasePlayerSDKInstence

Syntax

int AdvDVP_ReleasePlayerSDKInstence(void *p)

Parameters

p:	The playback SDK instance is created by "AdvDVP_CreatePlayerSDKInstence" function.

Return Value

SUCCEEDED:	Function succeeded.

Description

This function releases playback SDK instance created by the "AdvDVP_CreatePlayerSDKInstence" function.

See Also

AdvDVP_CreatePlayerSDKInstence

AdvDVP_OpenFile

Syntax

int AdvDVP_OpenFile(LPCSTR lpcsFileName)

Parameters

lpcsFileName:	Specifies the file name of the source video file.

Return Value

PLAYER_SUCCEEDED:	Function succeeded.
PLAYER_FAILED:	Function failed.

Description

This function opens the source video file and initializes the video player. The playback status would be set as “PLAYER_STOPPED” after successfully calling this function.

See Also

AdvDVP_CloseFile
AdvDVP_GetStatus

AdvDVP_CloseFile

Syntax

int AdvDVP_CloseFile()

Parameters

None.

Return Value

PLAYER_SUCCEEDED:	Function succeeded.
PLAYER_FAILED:	Function failed.

Description

This function closes the source video file and free resources allocated for video player. The playback status would be set as “PLAYER_NOTOPENED” after successfully calling this function.

See Also

AdvDVP_OpenFile
AdvDVP_GetStatus

AdvDVP_Play

Syntax

int AdvDVP_Play(HWND hwndApp, BOOL bAutoResizeWnd)

Parameters

hwndApp:	A windows handle for display area.
bAutoResizeWnd:	Specifies if the display area is resized automatically according to the video resolution.

Return Value

PLAYER_SUCCEEDED:	Function succeeded.
PLAYER_FAILED:	Function failed.

Description

This function plays the file that has been opened. The playback status would be set as “PLAYER_PLAYING” after successfully calling this function.

See Also

AdvDVP_Pause
AdvDVP_Stop
AdvDVP_GetStatus

AdvDVP_Pause

Syntax
int AdvDVP_Pause()

Parameters
None.

Return Value

PLAYER_SUCCEEDED:	Function succeeded.
PLAYER_FAILED:	Function failed.

Description
This function pauses or continues the file that has been opened. The playback status would be set as "PLAYER_PAUSED" after successfully calling this function.

See Also
AdvDVP_Play
AdvDVP_Stop
AdvDVP_GetStatus

AdvDVP_Stop

Syntax

```
int AdvDVP_Stop()
```

Parameters

None.

Return Value

PLAYER_SUCCEEDED:	Function succeeded.
PLAYER_FAILED:	Function failed.

Description

This function stops the file that is playing. The playback status would be set as “PLAYER_STOPPED” after successfully calling this function.

See Also

AdvDVP_Play
AdvDVP_Pause
AdvDVP_GetStatus

AdvDVP_Fast

Syntax
int AdvDVP_Fast()

Parameters
None.

Return Value

PLAYER_SUCCEEDED:	Function succeeded.
PLAYER_FAILED:	Function failed.

Description
This function improves the current play speed by one time, 4 times at most. The playback status would be set as "PLAYER_PLAYING" after successfully calling this function.

See Also
AdvDVP_Pause
AdvDVP_Stop
AdvDVP_Slow
AdvDVP_GetStatus

AdvDVP_Slow

Syntax
int AdvDVP_Slow()

Parameters
None.

Return Value

PLAYER_SUCCEEDED:	Function succeeded.
PLAYER_FAILED:	Function failed.

Description
This function slows the current play speed by one time, 4 times at most. The playback status would be set as "PLAYER_PLAYING" after successfully calling this function.

See Also
AdvDVP_Pause
AdvDVP_Stop
AdvDVP_Fast
AdvDVP_GetStatus

AdvDVP_PlayStep

Syntax

int AdvDVP_PlayStep()

Parameters

None.

Return Value

PLAYER_SUCCEEDED:	Function succeeded.
PLAYER_FAILED:	Function failed.

Description

This function makes the video to step forward one frame. The playback status would be set as "PLAYER_PAUSED" after successfully calling this function.

See Also

AdvDVP_Pause
AdvDVP_Stop
AdvDVP_GetStatus

AdvDVP_GetStatus

Syntax

int AdvDVP_GetStatus ()

Parameters

None

Return Value

PLAYER_SUCCEEDED:	Function succeeded.
PLAYER_FAILED:	Function failed.

Description

This function gets playback status.

```
typedef enum tagPlayerStatus{
    PLAYER_NOTOPENED = 0,
    PLAYER_OPENED      = 1,
    PLAYER_PLAYING     = 2,
    PLAYER_STOPPED     = 3,
    PLAYER_PAUSED      = 4
} PlayerStatus;
```

See Also

AdvDVP_OpenFile
AdvDVP_CloseFile
AdvDVP_Play
AdvDVP_Pause
AdvDVP_Stop
AdvDVP_Fast
AdvDVP_Slow
AdvDVP_PlayStep

AdvDVP_GetCurImage

Syntax

int AdvDVP_GetCurImage(LPBYTE *lpImage,
long *pBufSize)

Parameters

lpImage:	A pointer to a image buffer.
pBufSize:	A long pointer to receive the returned image buffer size.

Return Value

PLAYER_SUCCEEDED:	Function succeeded.
PLAYER_FAILED:	Function failed.

Description

This function gets current played image.

See Also

AdvDVP_RegNotifyMsg

Syntax

int AdvDVP_RegNotifyMsg(HWND hWnd, UINT nMsg)

Parameters

hWnd:	Specifies the handle of the window receiving this message.
nMsg:	Specifies the user-define message. When this message is received, it means some event of the playback occur such as the file playing is end.

Return Value

PLAYER_SUCCEEDED:	Function succeeded.
PLAYER_FAILED:	Function failed.

Description

This function registers a user-define message. When an event of the playback occurs, this message will be sent to the specified window.

This function must be called after "AdvDVP_OpenFile" function.

See Also

AdvDVP_CheckFileEnd

AdvDVP_CheckFileEnd

Syntax

BOOL AdvDVP_CheckFileEnd ()

Parameters

None

Return Value

If the event that the file playing end is detected, this function returns TRUE. Otherwise, the function returns FALSE.

Description

This function checks if the file playing is end.

See Also

AdvDVP_RegNotifyMsg

AdvDVP_GetVideoResolution

Syntax

int AdvDVP_GetVideoResolution(int *nWidth, int *nHeight)

Parameters

nWidth:	An integer pointer to get the width of the video.
nHeight:	An integer pointer to get the height of the video.

Return Value

PLAYER_SUCCEEDED:	Function succeeded.
PLAYER_FAILED:	Function failed.

Description

This function gets width and the height of the video.

See Also

AdvDVP_GetPlayRate

Syntax
double AdvDVP_GetPlayRate()

Parameters
None

Return Value
If the function succeeded, the playback ratio is returned. Otherwise, the function returns 0.

Description
This function retrieves the playback rate.

See Also
AdvDVP_Play
AdvDVP_Fast
AdvDVP_Slow

AdvDVP_GetFileTime

Syntax

double AdvDVP_GetFileTime()

Parameters

None

Return Value

If the function succeeded, the total file time is returned. Otherwise, the function returns 0.

Description

This function retrieves total file time in seconds.

See Also

AdvDVP_GetPlayedTime
AdvDVP_SetPlayPosition

AdvDVP_GetPlayedTime

Syntax

```
double  AdvDVP_GetPlayedTime()
```

Parameters

None

Return Value

If the function succeeded, the current file time is returned. Otherwise, the function returns 0.

Description

This function retrieves current file time in seconds.

See Also

AdvDVP_GetFileTime
AdvDVP_SetPlayPosition

AdvDVP_SetPlayPosition

Syntax

```
int  AdvDVP_SetPlayPosition (double dTime)
```

Parameters

dTime:	Specifies the file time in seconds.

Return Value

PLAYER_SUCCEEDED:	Function succeeded.
PLAYER_FAILED:	Function failed.

Description

This function seeks the file to the specified file time.

See Also

AdvDVP_GetFileTime
AdvDVP_GetPlayedTime

AdvDVP_GetFileTotalFrames

Syntax

LONGLONG AdvDVP_GetFileTotalFrames()

Parameters

None

Return Value

If the function succeeded, the total number of the frame of the file is returned. Otherwise, the function returns 0.

Description

This function retrieves total number of the frame of the file.

See Also

AdvDVP_GetPlayedFrames

AdvDVP_GetPlayedFrames

Syntax

LONGLONG AdvDVP_GetPlayedFrames()

Parameters

None

Return Value

If the function succeeded, the current frame number of the file is returned. Otherwise, the function returns 0.

Description

This function retrieves current frame number of the file.

See Also

AdvDVP_GetFileTotalFrames