

# **Neusys Technology Inc.**

## **Nuvis-3304af Series**

***Intel® 3<sup>rd</sup>-Gen Core™ i7/i5 Fanless Vision System  
with PoE & Deterministic Trigger I/O***

## **User's Manual**

**Rev. A1**

Published October 29<sup>th</sup>, 2013



## Contents

Disclaimer.....	5
Declaration of Conformity .....	5
FCC .....	5
CE.....	5
Copyright and Trademarks .....	5
Chapter 1 Introduction.....	6
1.1 Overview.....	6
1.2 Product Specification.....	7
1.2.1 Specification of Nuvis-3304af .....	7
1.2.2 Specification of Optional Isolated DIO .....	8
1.3 Supported CPU List .....	9
Chapter 2 Getting to know your Nuvis-3304af .....	10
2.1 Unpacking your Nuvis-3304af .....	10
2.2 Front Panel I/O Functions .....	11
2.2.1 Power Button.....	11
2.2.2 Reset Button.....	12
2.2.3 LED Indicators.....	12
2.2.4 CFast Socket.....	13
2.2.5 PS/2 Keyboard and Mouse Connectors .....	13
2.2.6 Gigabit Ethernet Port.....	14
2.2.7 VGA Connector .....	15
2.2.8 Remote On/Off Control and Status LED Output .....	15
2.2.9 USB 3.0 Connectors.....	17
2.2.10 DVI/HDMI Connectors .....	17
2.2.11 USB 2.0 Connectors .....	18
2.2.12 Speaker-out and MIC-in Audio Jacks.....	19
2.3 Back Panel I/O Functions.....	20
2.3.1 4-Pin Mini-DIN Power Connector.....	20
2.3.2 3-Pin Terminal Block for DC Input.....	21
2.3.3 COM Ports (COM1 & COM2) .....	21
2.3.4 Gigabit 802.3af PoE Ports .....	22
2.3.5 Deterministic Trigger I/O.....	24
2.4 Internal I/O Functions.....	27
2.4.1 DDR3 SODIMM Sockets .....	27
2.4.2 Internal SATA#1 Port.....	28
2.4.3 Mini PCI Express Connector#1 (with SIM Socket).....	29
2.4.4 Mini PCI Express Connector#2 .....	30
2.4.5 Internal USB 2.0 Ports.....	31



2.5	Expansion Cassette .....	33
2.5.1	Cassette of Nuvis-3304af-E.....	34
2.5.2	Cassette of Nuvis-3304afP.....	35
2.5.3	Fan Option of Cassette.....	36
2.6	Mechanical Dimension .....	37
2.6.1	Top View of Nuvis-3304af.....	37
2.6.2	Front View of Nuvis-3304af .....	38
2.6.3	Side View of Nuvis-3304af .....	38
2.6.4	Bottom View of Nuvis-3304af .....	39
Chapter 3	Getting Start.....	40
3.1	Install DDR3 SODIMM Modules.....	40
3.2	Install a 2.5" HDD/SSD.....	41
3.3	Install an Add-on Card into Cassette .....	43
3.4	Mount your Nuvis-3304af .....	45
3.5	Connect DC power to you Nuvis-3304af.....	48
3.5.1	Connect DC Power via 4-pin Power Connector.....	48
3.5.2	Connect DC Power via 3-pin Pluggable Terminal Block .....	49
3.6	Power on your Nuvis-3304af.....	50
3.6.1	Power on Nuvis-3304af Using the Power Button.....	50
3.6.2	Power on Nuvis-3304af Using an External Non-latched Switch .....	50
3.6.3	Power on Nuvis-3304af Using Wake-on-LAN Function .....	51
Chapter 4	BIOS and Driver .....	55
4.1	BIOS Settings.....	55
4.1.1	COM1 & COM2 Operating Mode.....	55
4.1.2	SATA Controller Mode .....	56
4.1.3	Power On after Power Failure Option.....	57
4.1.4	Wake-on-LAN Option.....	57
4.1.5	Watchdog Timer for Booting .....	58
4.1.6	Select a Boot Device .....	59
4.1.7	Position New Boot Device .....	60
4.2	Operating System Support.....	62
4.3	Driver Installation .....	63
4.3.1	Install All Drivers Using "One-Click" Driver Installation .....	63
4.3.2	Install Drivers Manually .....	64
4.3.3	Install Driver for DTIO and Per-port PoE On/Off Control .....	65
Appendix A	Using Watchdog Timer & DTIO.....	66
	Install WDT and DIO Library .....	67
	Using WDT Function.....	69
	WDT Function Reference.....	69
	Using DIO Function .....	71



Wiring for DIO .....	71
DIO Channel Definition (Nuvis-3304af) .....	72
DIO Function Reference (Polling Mode) .....	73
InitDIO .....	73
DIReadLine .....	73
DIReadPort .....	73
DOWriteLine .....	74
DOWritePort .....	74
COS Function Reference .....	76
SetupDICOS .....	76
RegisterCallbackDICOS .....	77
StartDICOS .....	77
StopDICOS .....	77
DI-COS Example .....	78
DTIO Function Reference .....	80
SetupDTIO .....	80
StartDTIO .....	82
StopDTIO .....	82
SetUnitDTIO .....	82
GetUnitDTIO .....	83
DTIO Example 1 -- Deterministic Timing between DI & DO .....	84
DTIO Example 2 -- Infinite Pulse Train .....	86
Deterministic Trigger Fan-Out Function Reference .....	89
SetupDTFO .....	89
StartDTFO .....	91
StopDTFO .....	91
SetUnitDTFO .....	91
GetUnitDTFO .....	92
DTFO Example 1 -- Deterministic Trigger Fan-Out .....	93
Appendix B Using Per-Port PoE On/Off Control .....	96
Per-Port PoE On/Off Control Function Reference .....	96
GetStatusPoEPort .....	96
EnablePoEPort .....	96
DisablePoEPort .....	97



# Disclaimer

This manual is intended to be used as a practical and informative guide only and is subject to change without prior notice. It does not represent commitment from Neousys Technology Inc. Neousys shall not be liable for direct, indirect, special, incidental, or consequential damages arising out of the use of the product or documentation, nor for any infringements upon the rights of third parties, which may result from such use.

# Declaration of Conformity

## FCC

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.

## CE

The product(s) described in this manual complies with all applicable European Union (CE) directives if it has a CE marking. For computer systems to remain CE compliant, only CE-compliant parts may be used. Maintaining CE compliance also requires proper cable and cabling techniques.

# Copyright and Trademarks

This document contains proprietary information protected by copyright. All rights are reserved. No part of this document may be reproduced by any mechanical, electronic, or other means in any form without prior written permission of the manufacturer.

Company/product names mentioned herein are used for identification purposes only and are trademarks and/or registered trademarks of their respective companies.



# Chapter 1 Introduction

## 1.1 Overview

Nuvis-3304af is a vision system dedicatedly designed for machine vision applications. Inheriting Neosys' proven fanless architecture and Power-over-Ethernet technology, Nuvis-3304af combines superb computing performance, integrated camera interfaces and great reliability in its compact chassis.

As accurate trigger/strobe control is crucial for vision applications, Neosys developed a new technology, Deterministic Trigger I/O, or DTIO, on Nuvis-3304af. Unlike legacy isolated DIO, this patent-pending DTIO technology allows users to program a deterministic timing correlation between input and output signals at a resolution of 25 microseconds. With DTIO, your vision system can have extremely precise control for proximity sensor input, strobe output and camera trigger.

Camera connectivity is another key for vision systems. In addition to integrated PoE and USB3 ports, Nuvis-3304af is provided with Neosys' patent Cassette\* design for PCIe/PCI expansion. By installing dedicated interface card, Nuvis-3304af can work with analog, 1394, Camera Link or CoaXPress camera. Or you can integrate a motion control card to fulfill a all-in-one inspection system.

Combining the quad-core CPU performance, PoE/USB3 camera interface, innovative DTIO and Cassette technology, Nuvis-3304af is the perfect platform for you vision application.



\* R.O.C. Patent No. M456527

## 1.2 Product Specification

### 1.2.1 Specification of Nuvis-3304af

<b>System Core</b>	
Processor	Supports the following CPU <ul style="list-style-type: none"> <li>● Intel® Core™ i7-3610QE (2.3/3.3 GHz, 6 MB cache)</li> <li>● Intel® Core™ i5-3610ME (2.7/3.3 GHz, 3 MB cache)</li> </ul>
Chipset	Intel® HM76 Platform Controller Hub
Graphics	Integrated Intel® HD Graphics 4000 Controller
Memory	2x 204-pin SO-DIMM sockets, up to 16 GB DDR3 1333/1600 MHz SDRAM
<b>I/O Interface</b>	
PoE	4x Gigabit IEEE 802.3af (15.4W) PoE ports by Intel® I210
Ethernet	1x Gigabit Ethernet port by Intel® I210
Video Port	1x DB-15 connector for analog RGB, supporting 2048x1536 resolution 2x DVI connectors for HDMI/DVI outputs, supporting 1920x1080 resolution
USB	4x USB 3.0 ports and 4x USB 2.0 ports
Serial Port	2x software-programmable RS-232/422/485 (COM1 & COM2)
KB/MS	1x 6-pin mini-DIN connector for PS/2 keyboard/mouse
Audio	1x mic-in and 1x speaker-out
<b>Storage Interface</b>	
SATA HDD	1x Internal SATA port for 2.5" HDD/SSD installation
CFast	1x CFast socket
<b>Expansion Bus</b>	
Mini PCI-E	1x internal mini PCI Express socket with USIM socket 1x internal mini PCI Express socket
PCI Express	1x PCIe x16 slot @ 8-lanes PCIE signals in Cassette (Nuvis-3304af-E only)
PCI	1x 33MHz/32-bit 5V PCI bus (Nuvis-3304af-P only)
<b>Power Supply &amp; Ignition Control</b>	
DC Input	1x 4-pin power connector for 8~25V DC input (for AC adapter) 1x 3-pin pluggable terminal block for 8~25V DC input (for direct DC wiring)
Max. Power Consumption	With Core™ i7-3610QE: 72.96W (3.84A@19V)* With Core™ i5-3610ME: 48.83W (2.57A@19V)*
Remote Ctrl. & Status Output	1x 10-pin (2x5) wafer connector for remote on/off control and status LED output
<b>Mechanical</b>	
Dimension	240 mm (W) x 225 mm (D) x 88 mm (H)



Weight	4.4 Kg (including 2.5" HDD and DDR3 SODIMM)
Mounting	Wall-mounting (Standard) or DIN-Rail mounting (optional)
<b>Environmental</b>	
Operating Temperature	-25°C ~ 70°C **/** (with i5-3610ME) -25°C ~ 60°C **/** (with i7-3610QM)
Storage Temperature	-40°C ~85°C
Humidity	10%~90% , non-condensing
Vibration	Operating, 5 Grms, 5-500 Hz, 3 Axes (w/ SSD, according to IEC60068-2-64)
Shock	Operating, 50 Grms, Half-sine 11 ms Duration (w/ SSD, according to IEC60068-2-27)
EMC	CE/FCC Class A, according to EN 55022 & EN 55024

\* Full-loading power consumption is measured with 16GB DDR3 memory and one 2.5" SATA SSD installed under the following conditions:

- 100% CPU package power
- Operating for all GbE ports
- Excluding power consumed by external PoE devices

\*\* The CPU loading for high-temperature test is applied using Intel® Thermal Analysis Tool. For detail testing criteria, please contact Neosys Technology

\*\*\* For sub-zero operating temperature, a wide temperature HDD drive or Solid State Disk (SSD) is required.

## 1.2.2 Specification of Optional Isolated DIO

<b>Isolated Digital Input</b>	
No. of Channel	8-CH Isolated Digital Input Channels
Logic Level	Logic High: 5 to 24V Logic Low: 0 to 1.5V
Isolated Voltage	2500 Vrms
Input Resistance	4.7kΩ
Operation Mode	Polling I/O with Change-of-State Interrupt, DTIO
<b>Isolated Digital Output</b>	
No. of Channel	8-CH Isolated Digital Output Channels
Sink Current (per channel)	100 mA (sustained loading) 250 mA (peak loading)
Isolated Voltage	2500 Vrms
Operation Mode	Polling, Change-of-State Interrupt
Output Type	Power MOSFET + Analog Device iCoupler®
Operation Mode	Polling I/O, DTIO





## 1.3 Supported CPU List

Nuvis-3304af series accepts a PGA-type Intel® 3<sup>rd</sup>-Gen i7/i5 processor via the rPGA988 CPU socket. In addition to i7-3610QE, i5-3610ME, you may also select other processor listed below according to your consideration of cost and performance.

- Intel® Core™ i7-3840QM Processor (8M Cache, up to 3.80 GHz)
- Intel® Core™ i7-3820QM Processor (8M Cache, up to 3.70 GHz)
- Intel® Core™ i7-3740QM Processor (6M Cache, up to 3.70 GHz)
- Intel® Core™ i7-3720QM Processor (6M Cache, up to 3.60 GHz)
- Intel® Core™ i7-3630QM Processor (6M Cache, up to 3.40 GHz)
- Intel® Core™ i7-3612QM Processor (6M Cache, up to 3.10 GHz)
- Intel® Core™ i7-3610QM Processor (6M Cache, up to 3.30 GHz)
- Intel® Core™ i7-3610QM Processor (6M Cache, up to 3.30 GHz)
- **Intel® Core™ i7-3610QE Processor (6M Cache, up to 3.30 GHz) \***
- Intel® Core™ i7-3540M Processor (4M Cache, up to 3.70 GHz)
- Intel® Core™ i7-3520M Processor (4M Cache, up to 3.60 GHz)
- **Intel® Core™ i5-3610ME Processor (3M Cache, up to 3.30 GHz) \***
- Intel® Core™ i5-3380M Processor (3M Cache, up to 3.60 GHz)
- Intel® Core™ i5-3360M Processor (3M Cache, up to 3.50 GHz)
- Intel® Core™ i5-3340M Processor (3M Cache, up to 3.40 GHz)
- Intel® Core™ i5-3320M Processor (3M Cache, up to 3.30 GHz)
- Intel® Core™ i5-3230M Processor (3M Cache, up to 3.20 GHz)
- Intel® Core™ i5-3210M Processor (3M Cache, up to 3.10 GHz)
- **Intel® Core™ i3-3120ME Processor (3M Cache, 2.40 GHz) \***
- Intel® Core™ i3-3120M Processor (3M Cache, 2.50 GHz)
- Intel® Core™ i3-3110M Processor (3M Cache, 2.40 GHz)
- Intel® Celeron® Processor 1020M (2M Cache, 2.10 GHz)
- **Intel® Celeron® Processor 1020E (2M Cache, 2.20 GHz) \***
- Intel® Celeron® Processor 1000M (2M Cache, 1.80 GHz)

*The processors with \* are listed in Intel® Embedded Roadmap and with a 7-year life cycle support (2013~2019).*



# Chapter 2 Getting to know your Nuvis-3304af

## 2.1 Unpacking your Nuvis-3304af

When you receive the package of Nuvis-3304af, please check immediately if the package contains all the items listed in the following table. If any item is missing or damaged, please contact your local dealer or Neosys Technology Inc. for further assistance.

Item	Description	Qty
1	Nuvis-3304af fanless vision system (According to the configuration you order, CPU/DDR3/HDD may be included. Please verify these items if necessary.)	1
2	Accessory box, which contains <ul style="list-style-type: none"> <li>● Neosys Drivers &amp; Utilities DVD</li> <li>● Wall-mounting bracket</li> <li>● M4 screws for wall-mounting bracket</li> <li>● Foot pad</li> <li>● 3-pin pluggable terminal block</li> <li>● HDD thermal pad for 2.5" HDD/SSD (if HDD is not installed)</li> </ul>	1 2 4 4 1 1



## 2.2 Front Panel I/O Functions

On Nuvis-3304af, plenty of I/O functions are provided on front panel and back panel so you can easily access them. Most common computer I/O functions are placed on the front panel. In this section, we'll illustrate each I/O function on the front panel.

### 2.2.1 Power Button



The power button is a non-latched switch with LED for ATX mode on/off operation. To turn on the Nuvis-3304af, press the power button and the blue LED is lighted up. To turn off the Nuvis-3304af, you can either issue a shutdown command in OS, or just simply press the power button. In case of system halts, you can press and hold the power button for 5 seconds to compulsorily shut down the system. Please note that a 5 seconds interval is kept by the system between two on/off operations (i.e. once turning off the system, you shall wait for 5 seconds to initiate another power-on operation).

## 2.2.2 Reset Button



The reset button is used to manually reset the system in case of any abnormal condition. To avoid unexpected operation, the reset button is hidden behind the front panel. You need to use a pin-like object to push the reset button.

## 2.2.3 LED Indicators



There are three LED indicators on the front panel: HDD, WDT and a non-marked LED (reserved for future usage). The descriptions of these three LED are listed in the following table.

Indicator	Color	Description
HDD	Red	Hard drive indicator, flashing when SATA hard drive is active.
WDT	Yellow	Watchdog timer indicator, flashing when watchdog timer is started.
AUX/IGN	Green	Not used for Nuvis-3304af.

## 2.2.4 CFast Socket



Nuvis-3304af provides a CFast socket on the front panel. CFast is a new interface for portable storage device. It shares a similar form-factor as CompactFlash but leverages Serial ATA interface to offer higher data transfer rate. The CFast socket on Nuvis-3304af is a push-push type. You can insert and push a CFast card to the end to install it, and push the installed CFast card again to eject it.

## 2.2.5 PS/2 Keyboard and Mouse Connectors



Support of legacy PS/2 keyboard and mouse on Nuvis-3304af is implemented using industrial-grade ITE8783 Super IO chip (-40 to 85°C). There is one 6-pin Mini-DIN connector on the front panel to provide PS/2 KB/MS signals. You shall need to use a Y-cable to connect PS/2 keyboard and mouse at the same time.

## 2.2.6 Gigabit Ethernet Port



Nuvis-3304af controller offers one GbE port and four Gigabit 802.3af PoE ports using Intel® I210 Gigabit Ethernet controllers. Each port has one dedicated GbE controller and one dedicated PCI Express link to present maximal network performance. The GbE ports is located on the front panel and it support Wake-on-LAN function (please refer to section 3.6.3 for detail). When plugging in the Ethernet cable, you can tell the Ethernet status and speed from the LED indicators on the RJ45 connector as following:

### Active/Link LED

LED Color	Status	Description
Yellow	Off	Ethernet port is disconnected
	On	Ethernet port is connected and no data transmission
	Flashing	Ethernet port is connected and data is transmitting/receiving

### Speed LED

LED Color	Status	Description
Green or Orange	Off	10 Mbps
	Green	100 Mbps
	Orange	1000 Mbps

To utilize the GbE port in Windows, you need to install corresponding driver for Intel® I210 GbE controller. Please refer to section 4.3.2 for information of driver installation.

## 2.2.7 VGA Connector

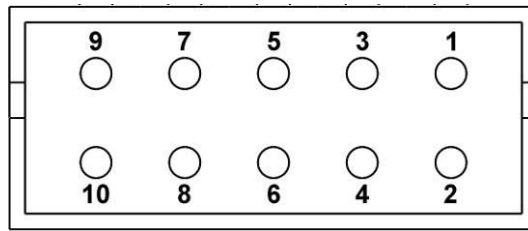


Nuvis-3304af has multiple display outputs on its front panel for connecting different displays according to your system configuration. VGA connector is the most popular way for connecting a display. The VGA output of Nuvis-3304af supports up to 2048 x 1536 resolution. By BIOS default and hardware implementation, the VGA output is always enabled in any case. To achieve best VGA output resolution in Windows, you need to install corresponding graphics driver. Please refer to section 4.3.2 for information of driver installation.

## 2.2.8 Remote On/Off Control and Status LED Output



For an application which places Nuvis-3304af inside a cabinet, it's useful to control the on/off of the system via an external switch, as well as check how the system's running via some external LED indicators. Nuvis-3304af provides a 2x5, 2.0mm pitch wafer connector on the front panel for this purpose.



Pin#	Definition	Description
1	Ctrl+	<b>[Input]</b> Remote on/off control, connecting to an external switch to turn on/off the system (polarity is negligible).
2	Ctrl-	
3	Power+	<b>[Output]</b> System power indicator, on if system is turned on, off if system is turned off.
4	Power-	
5	HDD+	<b>[Output]</b> Hard drive indicator, flashing when SATA hard drive is active.
6	HDD-	
7	IGN+	<b>[Output]</b> Not used on Nuvis-3304af
8	IGN-	
9	WDT+	<b>[Output]</b> Watchdog timer indicator, flashing when watchdog timer is started.
10	WDT-	

### Note

Please make sure the polarity is correct when you connect the external LED indicator to the Status LED Output.

Pin#1 and pin#2 are used to turn on or turn off the system remotely by connecting to an external switch. Users should connect a non-latched switch to Ctrl+/Ctrl- as it acts exactly the same as the power button on the front panel. For detail information of using remote on/off control function, please refer to section 3.6.2.

Pin#3 to pin#10 are used to output the system status including power, HDD, ignition signal and watchdog timer status. The status LED output has a built-in series-resistor and provides 3.3V, 10mA current, which means you can use these pins to directly drive an external LED indicator.



## 2.2.9 USB 3.0 Connectors



Nuvis-3304af offers four USB 3.0 (SuperSpeed USB) ports on its front panel. By BIOS default, these USB ports are operated in xHCI (eXtensible Host Controller Interface) mode and are compatible with USB 3.0, USB 2.0, USB 1.1 and USB 1.0 devices. Legacy USB support is also provided so you can use USB keyboard/mouse in DOS environment. To use USB 3.0 ports in Windows 7, you need to install USB 3.0 driver. Please refer to section 4.3.2 for information of driver installation.

### Note

1. *Intel USB 3.0 driver does not support Windows XP. In Windows XP, all USB 3.0 ports will work in USB 2.0 mode.*

## 2.2.10 DVI/HDMI Connectors



Nuvis-3304af has multiple display outputs on its front panel for connecting different displays according to your system configuration. DVI/HDMI transmits graphics data in digital format and therefore can deliver better image quality at high resolution. Two DVI/HDMI connectors on the front panel can either output DVI signals or HDMI signal depending on the display device connected. They support up to 1920x1200 resolution. You shall need a DVI to HDMI cable when connecting to a HDMI display device.



To achieve best DVI/HDMI output resolution in Windows, you need to install corresponding graphics driver. Please refer to section 4.3.2 for information of driver installation.

## 2.2.11 USB 2.0 Connectors



In addition to four USB 3.0 ports, Nuvis-3304af provides another four USB 2.0 ports on the front panel. By BIOS default, these USB 2.0 ports are operated in EHCI (Enhanced Host Control Interface) mode and are compatible with USB 2.0, USB 1.1 and USB 1.0 devices. Legacy USB support is provided so you can use USB keyboard/mouse in DOS environment.

## 2.2.12 Speaker-out and MIC-in Audio Jacks



Nuvis-3304af provides audio function using Intel® High Definition Audio (built-in in HM76 PCH) and Realtek ALC262 codec. There are two audio jacks on the front panel. The pink one is used for microphone input, and the green one is used for speaker output. To utilize the audio function in Windows, you need to install corresponding drivers for both Intel® HM76 PCH chipset and Realtek ALC262 codec. Please refer to section 4.3.2 for information of driver installation.

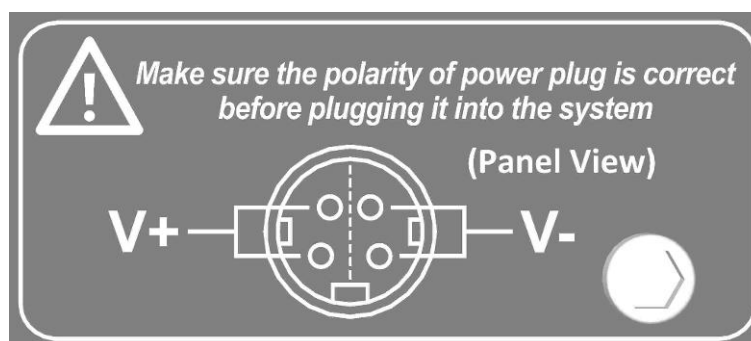
## 2.3 Back Panel I/O Functions

To fit more general application requirements, Nuvis-3304af offers more I/O functions on its back panel. In this section, we'll illustrate each I/O function on the back panel.

### 2.3.1 4-Pin Mini-DIN Power Connector



Nuvis-3304af allows a wide range of DC power input from 8 to 25V. It offers two ways for connecting DC power: a 4-pin mini-DIN power connector or a 3-pin pluggable terminal block. The 4-pin mini-DIN power connector is used to connect the power plug of an AC/DC adapter. It's convenient for indoor usage where AC power is usually available. Since there is no specific rule of pin definition for this type of connector, please always confirm the polarity of the power connector in prior to plug it into Nuvis-3304af if you're not using the power adapter provided by Neousys.



#### Caution

1. Please make sure the voltage of DC power is correct before you connect it to Nuvis-3304af. Supplying a voltage over 25V will damage the system.
2. You should use either 4-pin mini-DIN power connector or 3-pin pluggable terminal block for DC power input. **DO NOT** supply power to both connectors at the same time.

## 2.3.2 3-Pin Terminal Block for DC Input



Nuvis-3304af allows a wide range of DC power input from 8 to 25V. It offers two ways for connecting DC power: a 4-pin mini-DIN power connector or a 3-pin pluggable terminal block. The 3-pin pluggable terminal block is fit for field usage where DC power is usually provided. And the screw clamping connection of terminal block gives a very reliable way of wiring the DC power. For detail information of supplying DC power via pluggable terminal block, please refer to section 3.5.2.

### Caution

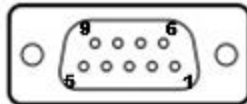
1. Please make sure the voltage of DC power is correct before you connect it to Nuvis-3304af. Supplying a voltage over 25V will damage the system.
2. You should use either 4-pin mini-DIN power connector or 3-pin pluggable terminal block for DC power input. **DO NOT** supply power to both connectors at the same time.

## 2.3.3 COM Ports (COM1 & COM2)



Nuvis-3304af provides two COM ports on the back panel for communicating with external devices. COM1 and COM2 are located on the back panel via 9-pin D-Sub male connectors. They are implemented using industrial-grade ITE8783 Super IO chip (-40 to 85°C) and provide up to 115200 bps baud rate.

Both COM1 and COM2 support software-selectable RS-232/422/485 mode. The operation mode of COM1/COM2 can be set in BIOS setup utility (refer to section 4.1.1 for detail). The following table describes the pin definition of COM ports.



Pin#	RS-232 Mode	RS-422 Mode	RS-485 Mode (Two-wire 485)
1	DCD		
2	RX	422 TXD+	485 TXD+/RXD+
3	TX	422 RXD+	
4	DTR	422 RXD-	
5	GND	GND	GND
6	DSR		
7	RTS		
8	CTS	422 TXD-	485 TXD-/RXD-
9	RI		

### 2.3.4 Gigabit 802.3af PoE Ports



PoE, or Power over Ethernet, is a technology to supply electrical power along with data on a standard CAT-5/CAT-6 Ethernet cable. Nuvis-3304af, acting as a PoE PSE (Power Sourcing Equipment), integrates four GigE PoE ports compliant with IEEE 802.3af standard.

Each PoE port can deliver 15.4 W of power to a PoE PD (Powered Device), such as a PoE camera. PoE defines a mechanism to automatically detect the device connected and determine whether to dispatch power. This makes PoE port 100% compatible with traditional Ethernet devices.

Four GigE PoE ports are located on the rear panel. These GigE ports are implemented using Intel® I210 Gigabit Ethernet controllers, which support features such as jumbo frame and link aggregation. Each port has one dedicated GigE controller and one dedicated PCI Express link to present maximal network performance. When plugging in the Ethernet cable, you can tell the Ethernet status and speed from the LED indicators on the RJ45 connector as following:

### Active/Link LED

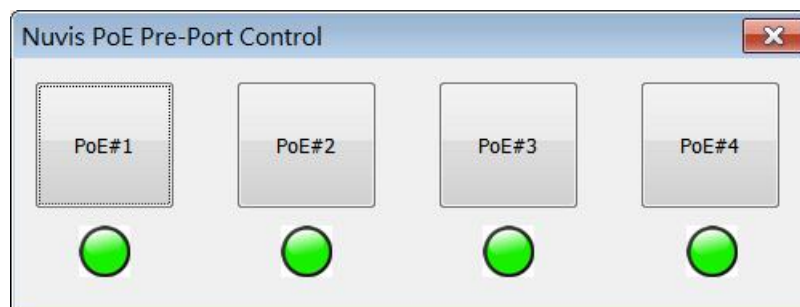
LED Color	Status	Description
Yellow	Off	Ethernet port is disconnected
	On	Ethernet port is connected and no data transmission
	Flashing	Ethernet port is connected and data is transmitting/receiving

### Speed LED

LED Color	Status	Description
Green or Orange	Off	10 Mbps
	Green	100 Mbps
	Orange	1000 Mbps

To utilize the GbE port in Windows, you need to install corresponding driver for Intel® I210 GbE controller. Please refer to section 4.3.2 for information of driver installation.

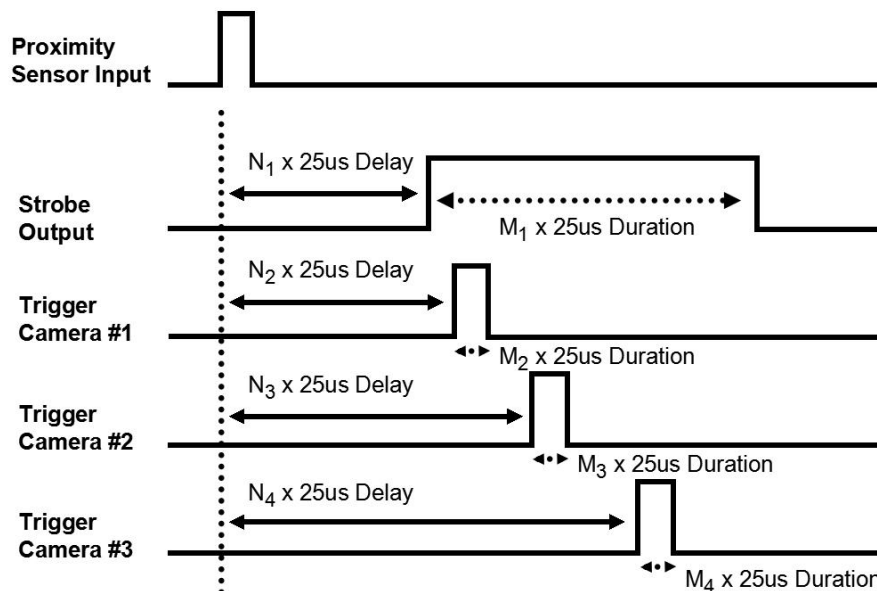
In addition to PoE PSE capability, Nuvis-3304af features unique per-port PoE power on/off control function. Via Neousys' driver API, users can program whether to deliver power to each PoE port. This can be useful for some fault-recovery or device power reset actions. Please refer to Appendix B for information about using the per-port power on/off control function.



## 2.3.5 Deterministic Trigger I/O



For machine vision applications, it is crucial to have accurate trigger/strobe control so clear image can be taken at the right position. To offer a deterministic timing correlation for trigger/strobe signals, a patent-pending technology called “Deterministic Trigger I/O”, or DTIO, is implemented on Nuvis-3304af. It’s a set of isolation digital I/O channels with a standalone MCU to allow a programmable and deterministic trigger/strobe behavior at a resolution of 25 microseconds.



The DTIO connector is located on the rear panel. It offers 8-CH isolated DI and 8-CH isolated DO. The following operating modes are supported:

### 1. Polling Read/Write with Change-of-State Interrupt

This is the typical way to access DIO function. Users can either get input status and program output status using polling scheme, or specify a input status (combination of





input channel(s) and rising/falling edge(s)) so that MCU will generate an interrupt accordingly. In this mode, it's user's liability to handle the interrupt and take corresponding actions.

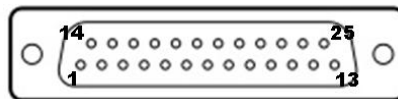
## 2. Deterministic Trigger I/O

In DTIO mode, MCU takes the responsibility of controlling input/output channels. Users can program to generate pulse output on one or multiple DO channels with deterministic delay and duration when a predefined trigger condition occurred (e.g. rising/falling edge on a designated DI channel). The built-in algorithm in MCU guarantees a 25 microseconds resolution for specifying the pulse delay and duration.

## 3. Trigger Fan-out

The trigger fan-out mode is a special scenario of DTIO. In some cases, users want to forward a trigger input to multiple output channels with minimal latency. In this mode, users can only specify the output duration. When a trigger condition occurred (i.e. rising/falling edge on a designated DI channel), pulses are generated immediately on designated channel(s).

For detailed information about programming the DTIO function, please refer to Appendix A Using Watchdog Timer & Isolated DIO. If you want to understand more about DTIO technology, please visit Neousys' website and download the technical white paper regarding DTIO.



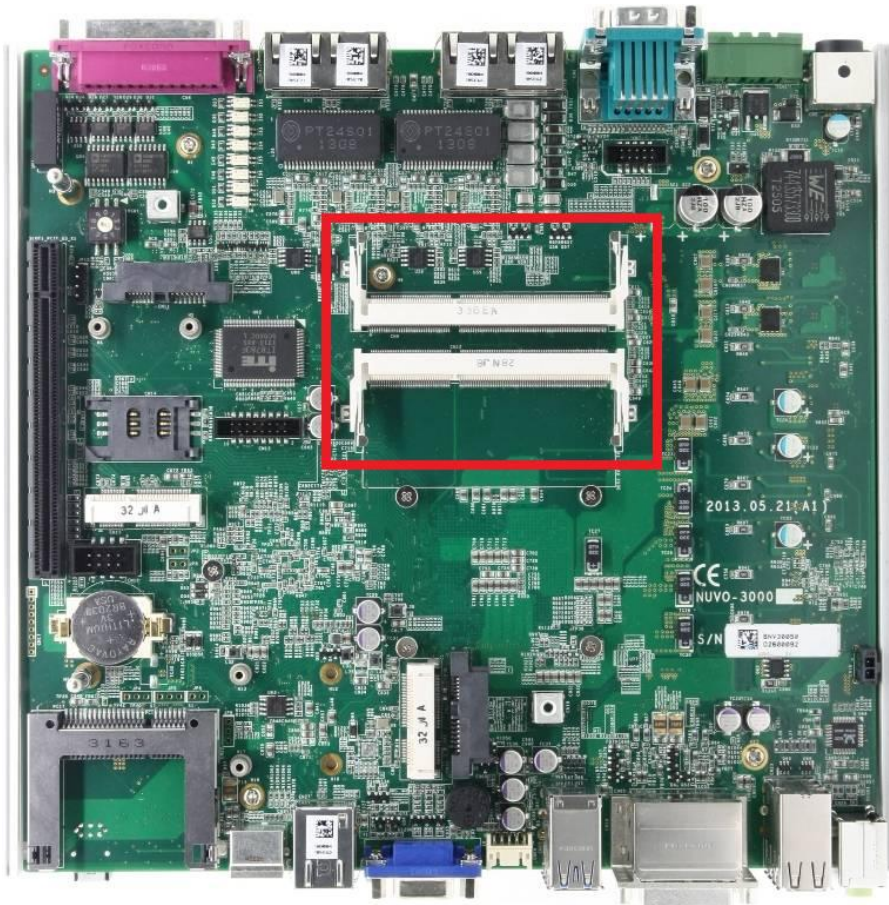
Pin#	Pin Definition	Pin#	Pin Definition
1	VDD	14	ISO_5V
2	DO_0	15	DO_GND
3	DO_3	16	DO_1
4	DO_GND	17	DO_2
5	DO_5	18	DO_4
6	DO_7	19	DO_6
7	DI_GND	20	DO_GND
8	DI_6	21	DI_GND
9	DI_1	22	DI_5
10	DI_GND	23	DI_3
11	DI_0	24	DI_7
12	DI_2	25	DI_GND
13	DI_4		



## 2.4 Internal I/O Functions

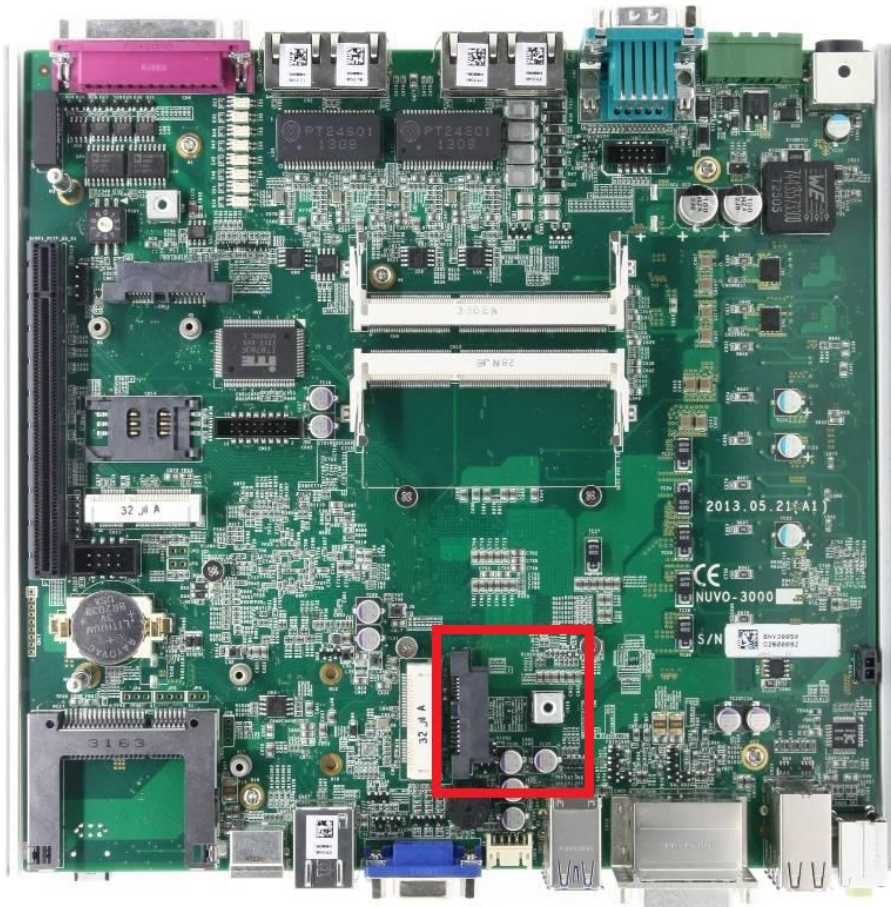
In addition to I/O connectors on the front/back panel, Nuvis-3304af provides other useful features via its on-board connectors, such as SATA ports, mini-PCIe sockets, internal USB ports, optional COM ports and etc. In this section, we'll illustrate these internal I/O functions.

### 2.4.1 DDR3 SODIMM Sockets



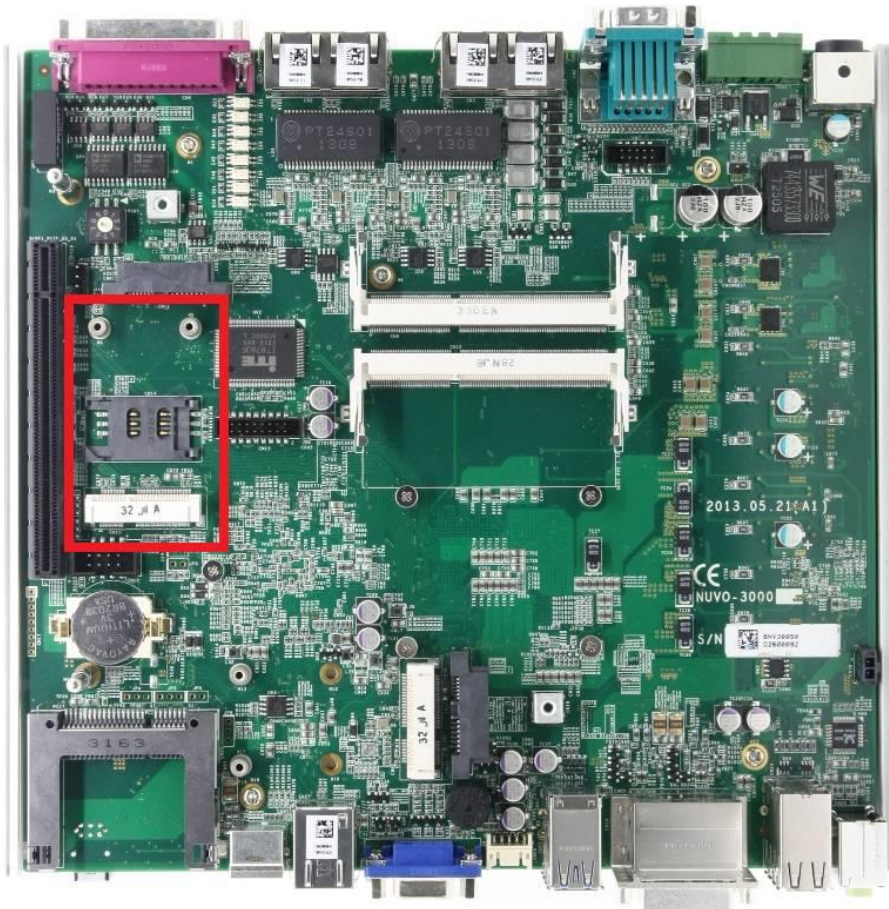
Nuvis-3304af provides two 204-pin, SODIMM sockets for installing DDR3 memory modules. It supports a maximal 16GB capacity by installing two 8GB DDR3 1600MHz SODIMM modules. For information of installing DDR3 memory modules, please refer to section 3.1 for detail.

## 2.4.2 Internal SATA#1 Port

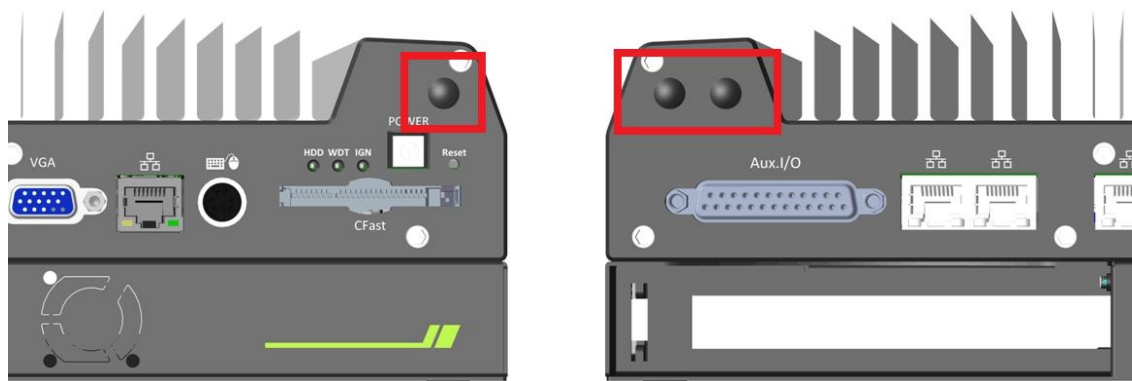


Nuvis-3304af provides one internal SATA port to accommodate one 2.5" SATA hard drive. SATA#1 Port is used in conjunction with the HDD bracket on "Pet-Door" to accommodate a 2.5" HDD/SSD. A special washer is installed to fix the SATA cable and make sure a very robust connection. For information of installing a HDD/SSD to SATA#1 port, please refer to section 3.2 for detail.

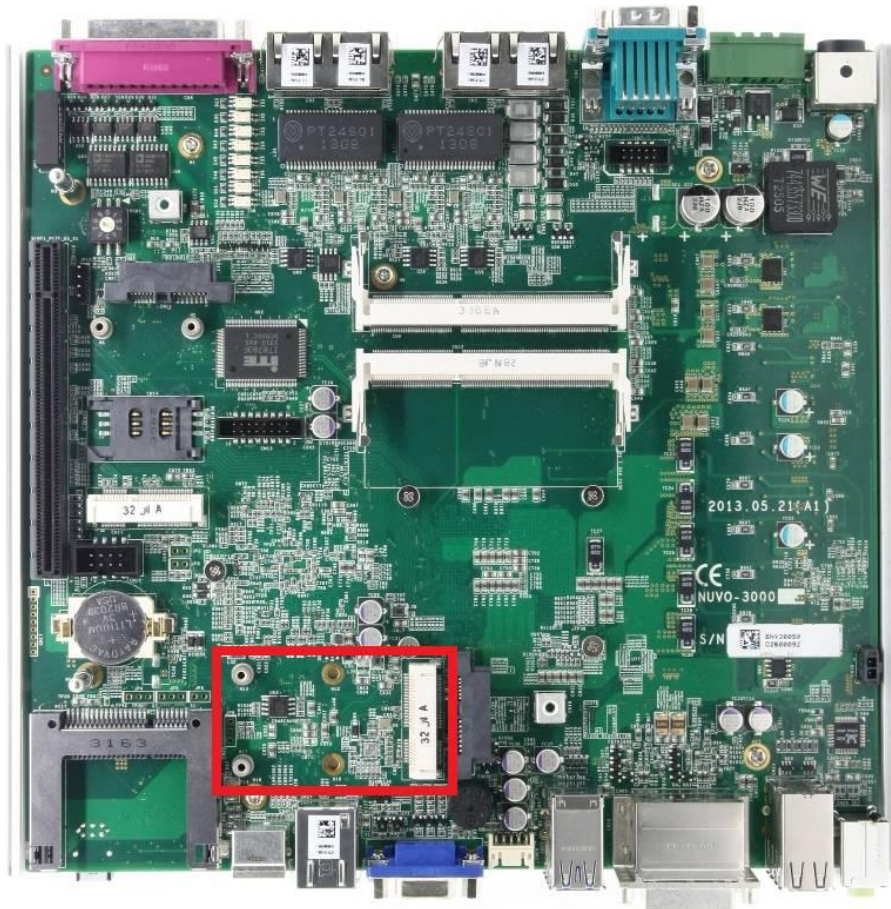
## 2.4.3 Mini PCI Express Connector#1 (with SIM Socket)



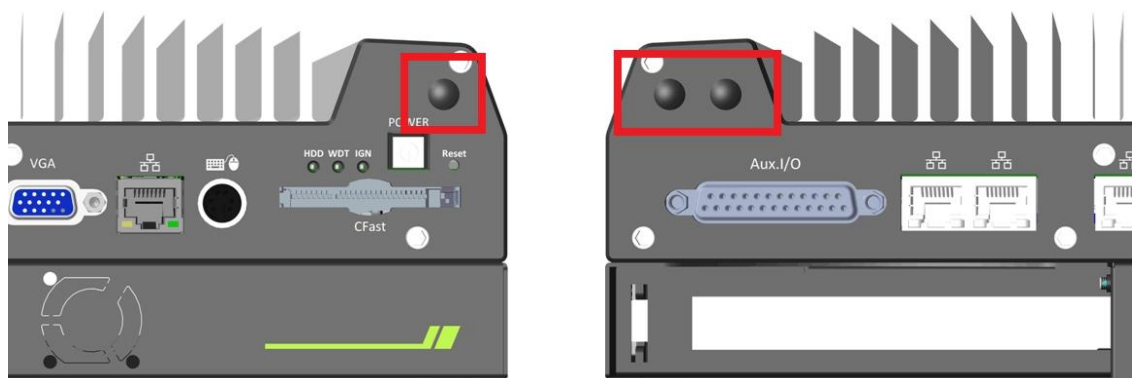
Nuvis-3304af provides two on-board Mini PCI Express connectors. There are plenty of off-the-shelf mini-PCIe modules with versatile capabilities. By installing a mini-PCIe module, your system can have expanded features such as WIFI, 3G, GPS, RAID and etc. Mini PCI Express Connector#1 is designed with SIM card support. With a SIM card installed, it's capable to connect your system to Internet in wide territory through telecom operator's GPRS/3G network. For WIFI/3G communication, Nuvis-3304af provides multiple SMA antenna apertures on the front and back panel for multi-antenna configuration.



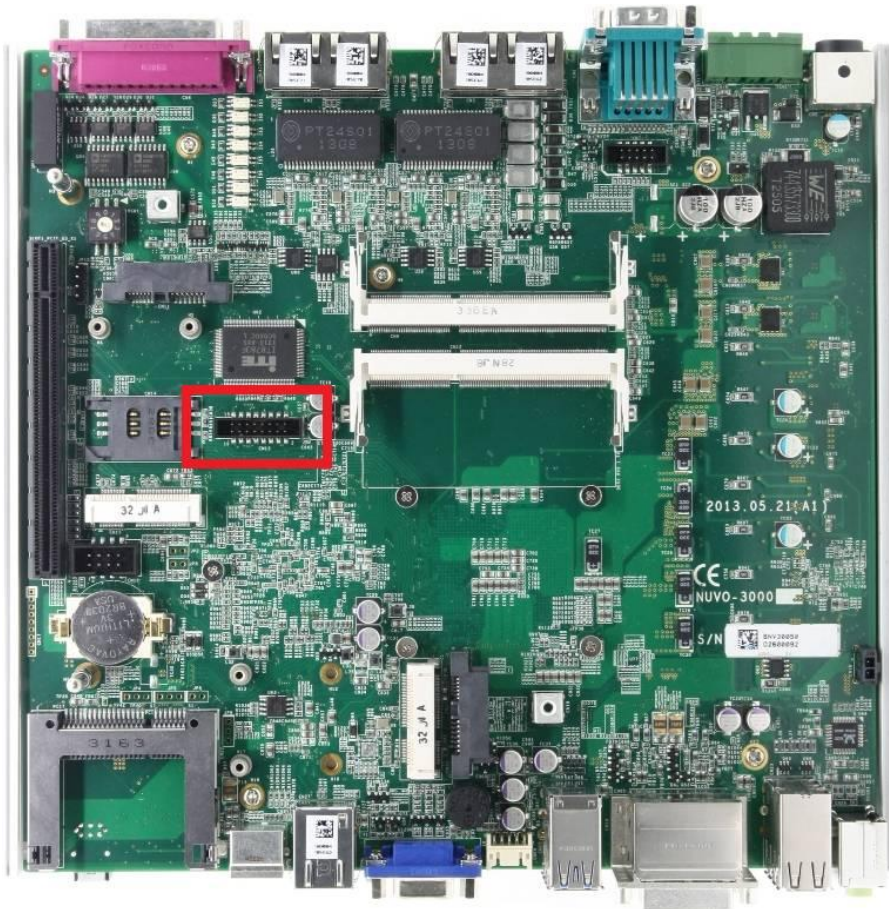
## 2.4.4 Mini PCI Express Connector#2



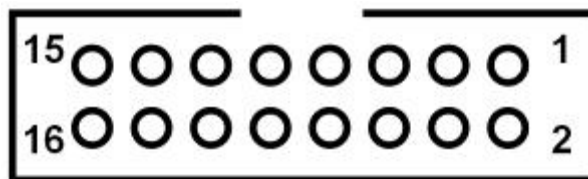
Nuvis-3304af provides two on-board Mini PCI Express connectors. There are plenty of off-the-shelf mini-PCIe modules with versatile capabilities. By installing a mini-PCIe module, your system can have expanded features such as WIFI, 3G, GPS, RAID and etc. Mini PCI Express Connector#2 is designed to accommodate full-size and half-size mini PCI-E module. You can install versatile modules, such as WIFI, GPS, RAID controller, mini PCI-E DOM and etc. For communication modules which need external antenna connection, Nuvis-3304af provides multiple SMA antenna apertures on the front and back panel for multi-antenna configuration.



## 2.4.5 Internal USB 2.0 Ports



Nuvis-3304af provides additional two USB 2.0 ports internally via a 2x8 pins, 2.0mm pitch box header. The internal USB ports are designed to allow users attaching a protection dongle inside the chassis. To use the internal USB port, you need a dedicated box-header to USB cable. Please contact Neosys for further information



Pin#	Definition	Description
1	VCC_USB	5V USB power
2	VCC_USB	5V USB power
3	USB_4_N	USB4 Differential data output - negative
4	USB_5_N	USB5 Differential data output - negative
5	USB_4_P	USB4 Differential data output - positive
6	USB_5_P	USB5 Differential data output - positive
7	GND	Ground

8	GND	Ground
9	VCC_USB	5V USB power
10	VCC_USB	5V USB power
11	N/A	N/A
12	N/A	N/A
13	N/A	N/A
14	N/A	N/A
15	GND	Ground
16	GND	Ground



## 2.5 Expansion Cassette

The expansion Cassette is an innovation design on Nuvis-3304af series. By providing a separated chassis to accommodate add-on card, it's more effective to manage the thermal conditions of both the system and the add-on card. The modular concept of Cassette also makes it easier to expand extra functions for your vision system.



Nuvis-3304af-E contains a Cassette with PCI Express slot, and Nuvis-3304af-P contains a Cassette with PCI slot. If users want to use an add-on card with higher power consumption, a fan option is offered by Neusys to create active air flow and maintain the stable thermal condition for Cassette.



## 2.5.1 Cassette of Nuvis-3304af-E

The expansion Cassette of Nuvis-3304afE contains a backplane with a x16 PCI Express connector. It runs 8-lanes, Gen2 PCI Express signals to provide a maximal 4GB/s bandwidth. Nuvis-3304afE supports 4A@12V rated current for a PCI Express add-on card with higher power consumption. You can also get the 12VDC from the on-board power connectors if necessary.





#	Connector	Function Description										
1	x16 PCI Express	x16 PCI Express connector that runs 8-lane, Gen2 signal. Compatible with x16, x8, x4 and x1 PCI Express add-on card.										
2	12VDC Fan Power	3-pin, 2.54mm pitch power connector for supplying 12VDC to the optional fan. <div style="text-align: center;">  </div> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Pin</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>N/A</td> </tr> <tr> <td>2</td> <td>12V (2A rated Current)</td> </tr> <tr> <td>3</td> <td>GND</td> </tr> </tbody> </table>	Pin	Description	1	N/A	2	12V (2A rated Current)	3	GND		
Pin	Description											
1	N/A											
2	12V (2A rated Current)											
3	GND											
3	SATA Signal	Standard 7-pin SATA connector										
4	5/12VDC Power	4-pin, 2.0mm pitch wafer connector for supplying 5/12VDC <div style="text-align: center;">  </div> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Pin</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>12V (2A rated Current)</td> </tr> <tr> <td>2</td> <td>GND</td> </tr> <tr> <td>3</td> <td>GND</td> </tr> <tr> <td>4</td> <td>5V (2A rated Current)</td> </tr> </tbody> </table>	Pin	Description	1	12V (2A rated Current)	2	GND	3	GND	4	5V (2A rated Current)
Pin	Description											
1	12V (2A rated Current)											
2	GND											
3	GND											
4	5V (2A rated Current)											

## 2.5.2 Cassette of Nuvis-3304afP

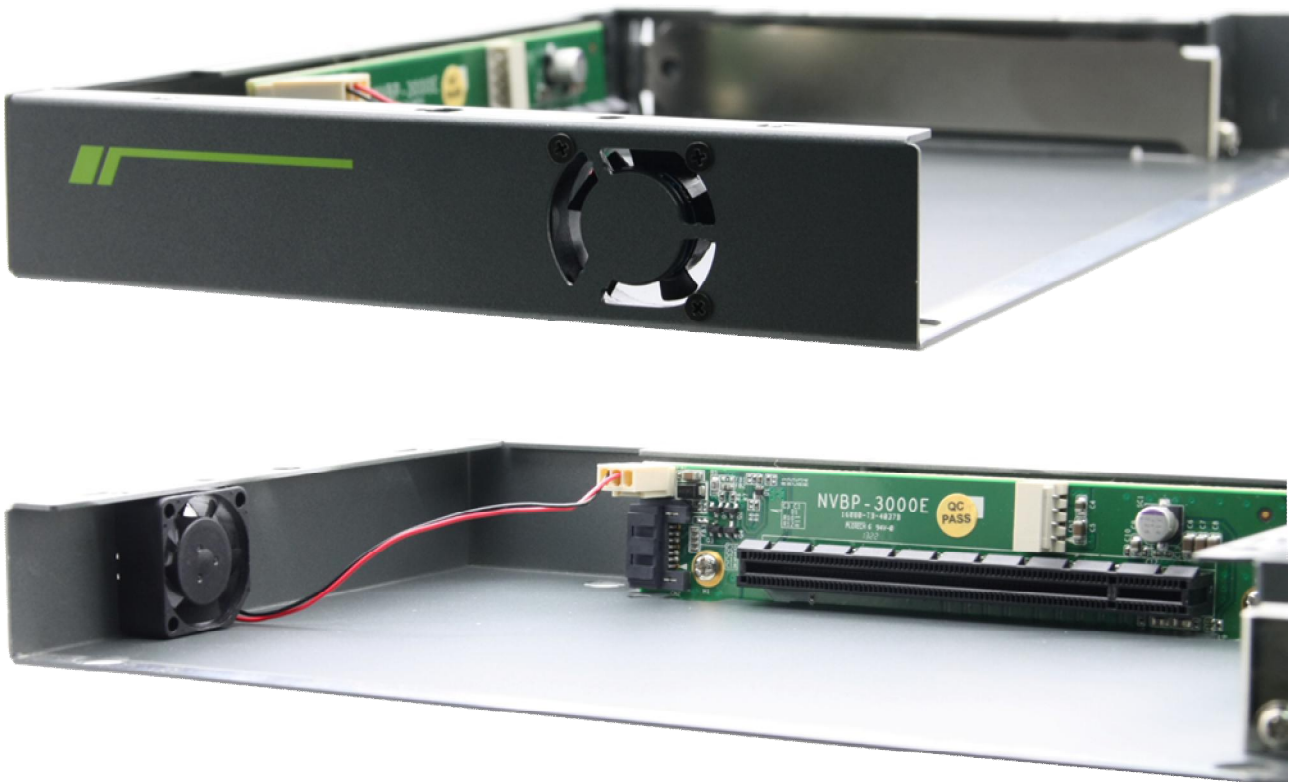
The expansion Cassette of Nuvis-3304afP contains a backplane with a 32-bit/33MHz PCI connector. Nuvis-3304afP supports 4A@12V rated current. You can get the 12VDC from the on-board power connectors if necessary.



#	Connector	Function Description										
1	33MHz/32-bit PCI	33MHz/32-bit PCI bus via PLX8112 PCIe-to-PCI bridge										
2	12VDC Fan Power	3-pin, 2.54mm pitch power connector for supplying 12VDC to the optional fan. <div style="text-align: center;">  <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Pin</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>N/A</td> </tr> <tr> <td>2</td> <td>12V (2A rated Current)</td> </tr> <tr> <td>3</td> <td>GND</td> </tr> </tbody> </table> </div>	Pin	Description	1	N/A	2	12V (2A rated Current)	3	GND		
Pin	Description											
1	N/A											
2	12V (2A rated Current)											
3	GND											
3	SATA Signal	Standard 7-pin SATA connector										
4	5/12VDC Power	4-pin, 2.0mm pitch wafer connector for supplying 5/12VDC <div style="text-align: center;">  <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Pin</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>12V (2A rated Current)</td> </tr> <tr> <td>2</td> <td>GND</td> </tr> <tr> <td>3</td> <td>GND</td> </tr> <tr> <td>4</td> <td>5V (2A rated Current)</td> </tr> </tbody> </table> </div>	Pin	Description	1	12V (2A rated Current)	2	GND	3	GND	4	5V (2A rated Current)
Pin	Description											
1	12V (2A rated Current)											
2	GND											
3	GND											
4	5V (2A rated Current)											

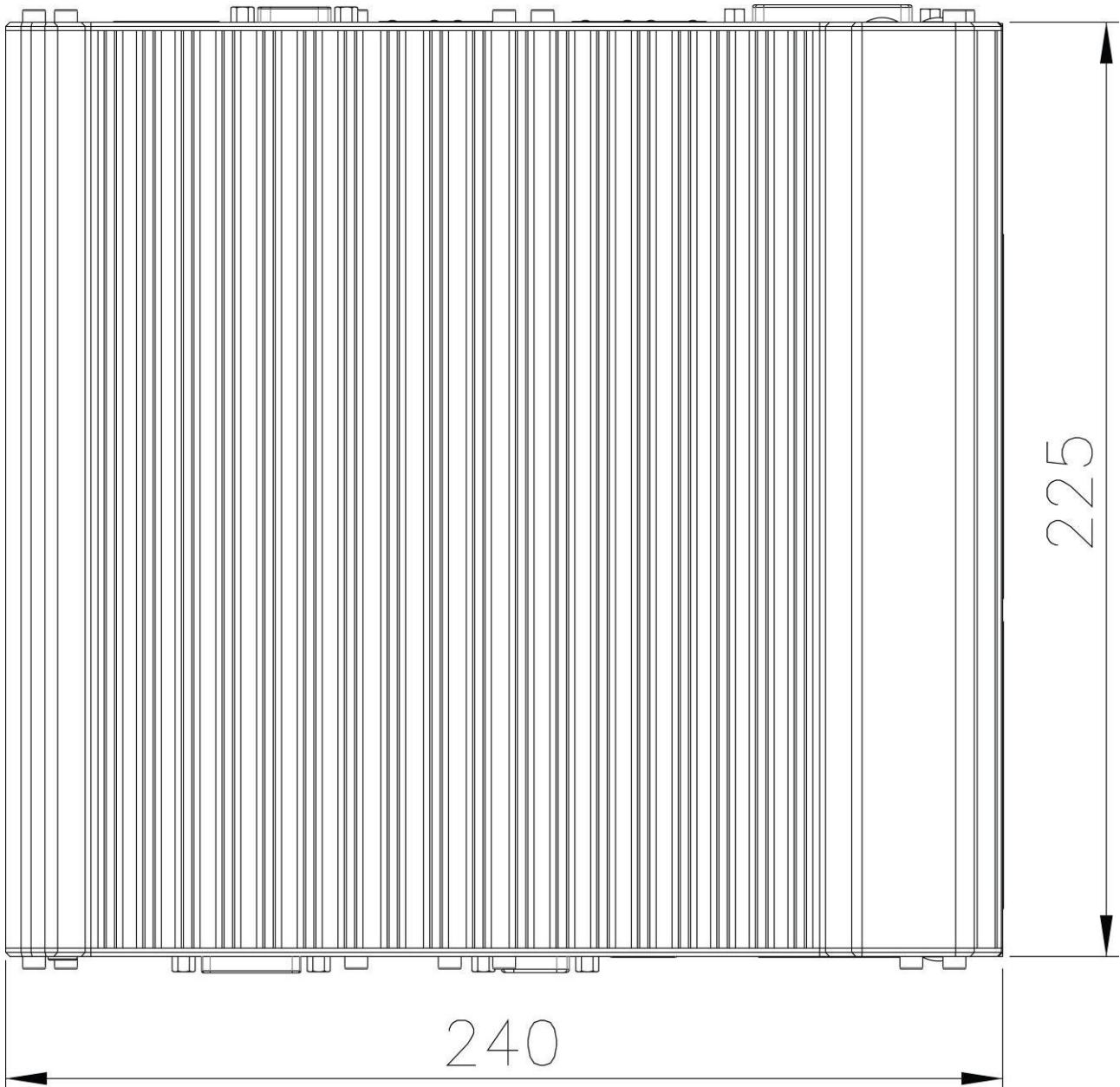
## 2.5.3 Fan Option of Cassette

When an add-on card with higher power consumption installed in Cassette, it is user's liability to consider the thermal dissipation. Neusys offers a fan option, as a general solution for versatile add-on cards, to create active air flow and maintain a proper ambient temperature inside Cassette. If conduction-cooling scheme is needed, a customized a heat-spreader shall be made accordingly to contact components on add-on card and the surface of Cassette.

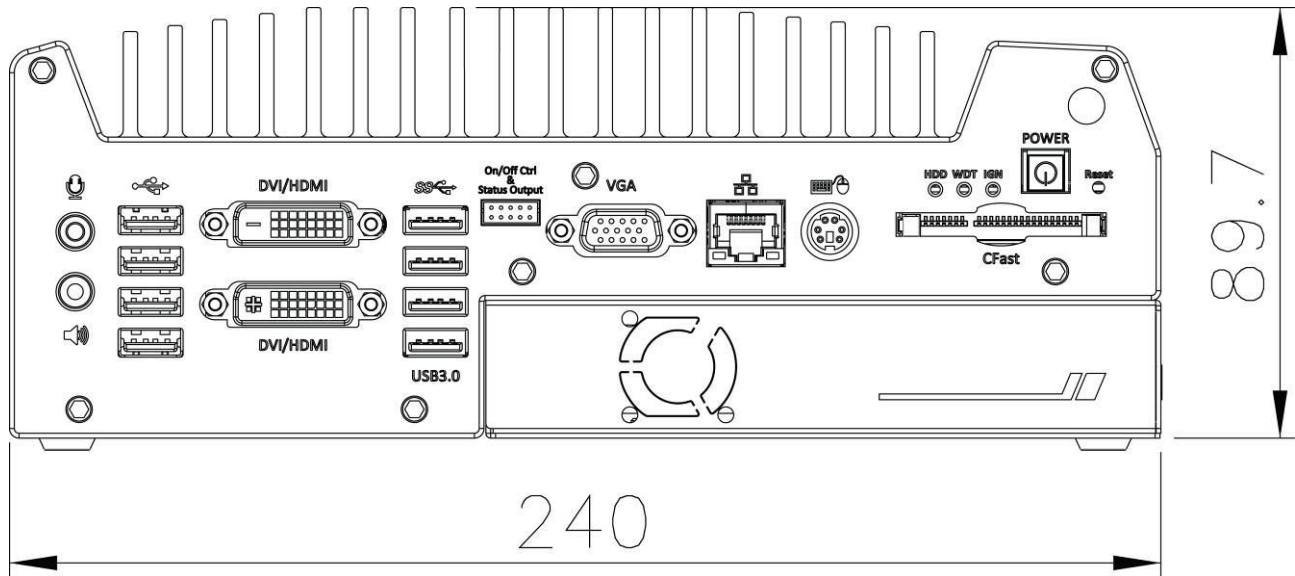


## 2.6 Mechanical Dimension

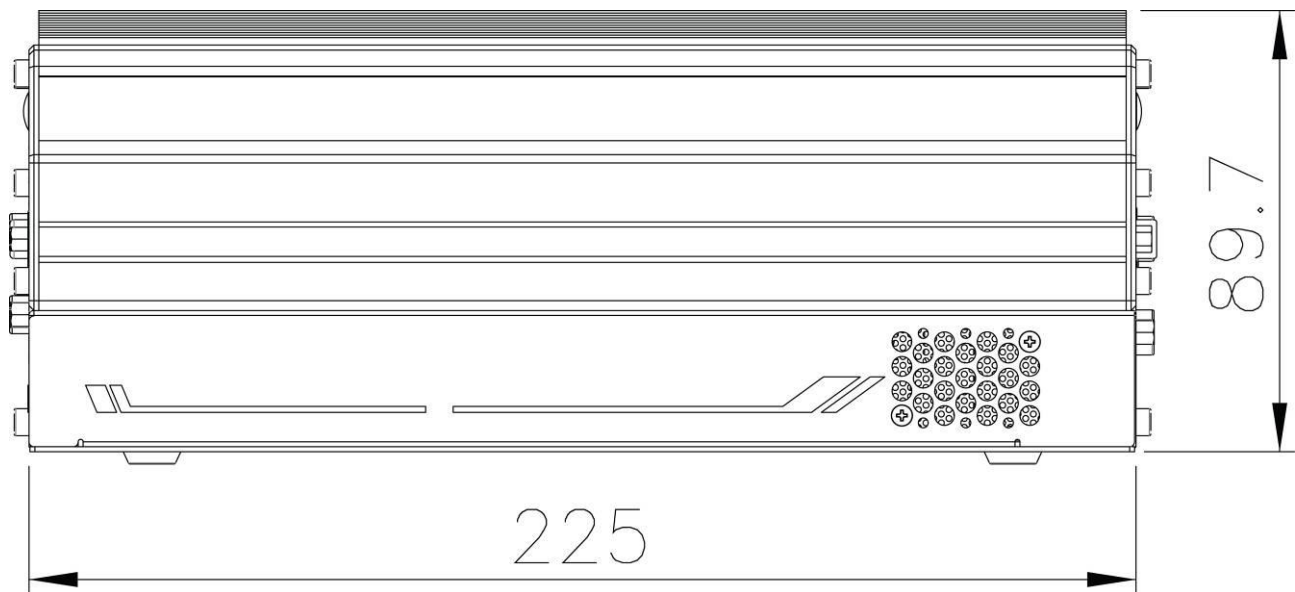
### 2.6.1 Top View of Nuvis-3304af



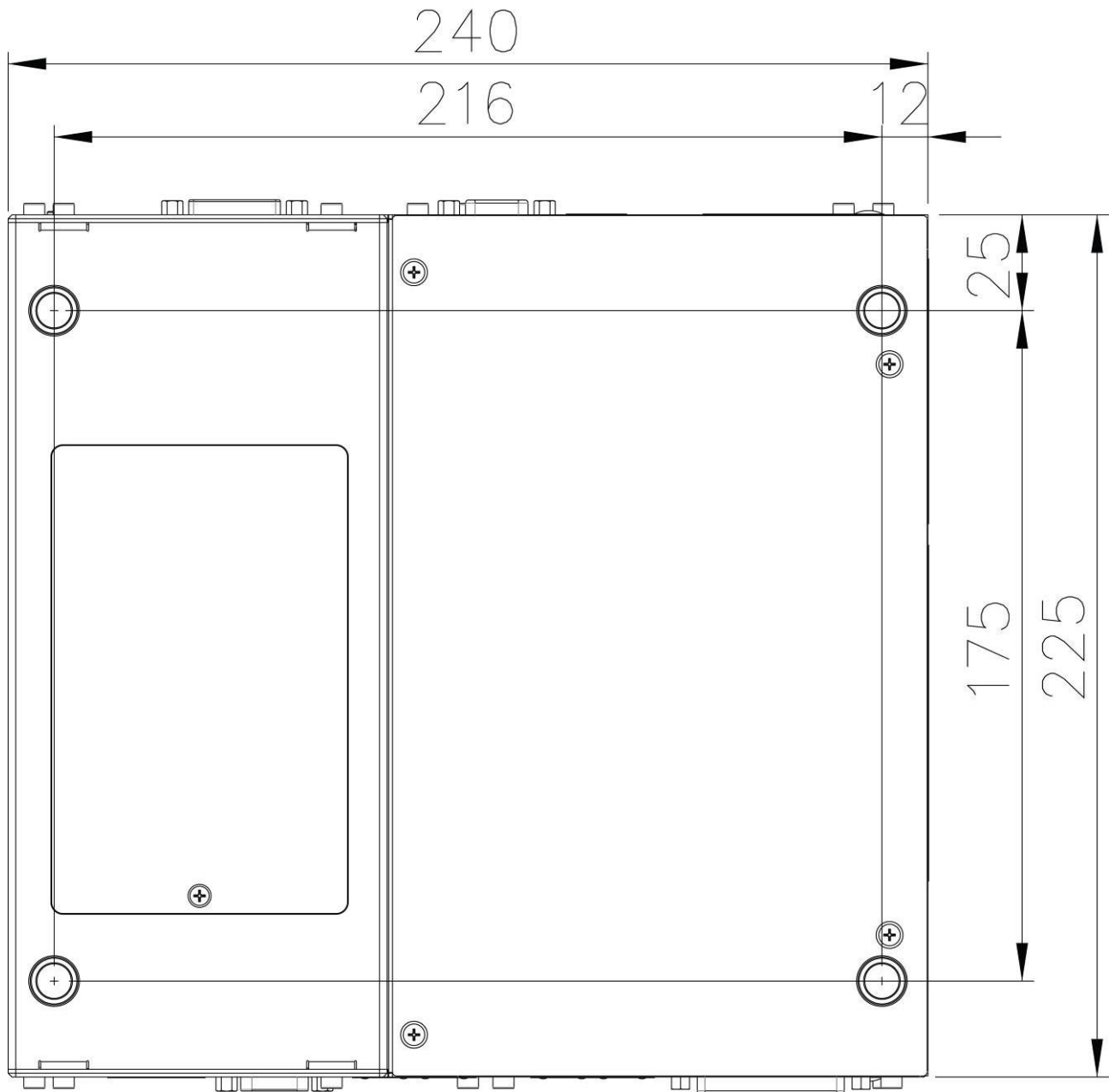
## 2.6.2 Front View of Nuvis-3304af



## 2.6.3 Side View of Nuvis-3304af



## 2.6.4 Bottom View of Nuvis-3304af



# Chapter 3 Getting Start

## 3.1 Install DDR3 SODIMM Modules

Nuvis-3304af provides two 204-pin, SODIMM sockets for installing DDR3 memory modules. It supports a maximal 16GB capacity by installing two 8GB DDR3 1600MHz SODIMM modules. You can install/replace DDR3 SODIMM modules by following the steps listed below.

1. Put the Nuvis-3304af upside down on a flat surface. You can see the “Pet-Door” exposed. Use a Philips screwdriver to loosen the M3 flat-head screw and open the “Pet-Door”.



2. Use a Philips screwdriver to loosen four M4 flat-head screws on the Cassette. Pull up the Cassette to remove it from the Nuvis-3304af.





3. Tile the SODIMM module and insert it to the SODIMM socket. As it's firmly contacted with socket connectors, press it down until the clamps of the socket snap into the latching position of SODIMM module. Repeat this step if you want to install the second DDR3 SODIMM module.



## 3.2 Install a 2.5" HDD/SSD

Internal SATA#1 port on Nuvis-3304af is used in conjunction with the HDD bracket on "Pet-Door" to accommodate a 2.5" HDD/SSD. You can install/replace the HDD/SSD attached to SATA#1 port by following the steps listed below.

1. Follow step#1 in section 3.1 to open the "Pet-Door". You can see a SATA cable exposed.
2. Find the HDD bracket come with the "Pet-Door", M3 flat-head screws (4 pieces), and a HDD thermal pad in the accessory box. Remove the films on both sides of the thermal pad and place it on the center of HDD bracket.



3. Place the HDD into the bracket and gently push it down to make it contact with the thermal pad. Use a Philips screwdriver to fix the HDD with M3 flat-head screws. Please note that the HDD must be placed in the right direction as below.



*(another 2 screws are on the opposite side)*

4. Pull out the SATA cable inside the chassis and connect it to HDD



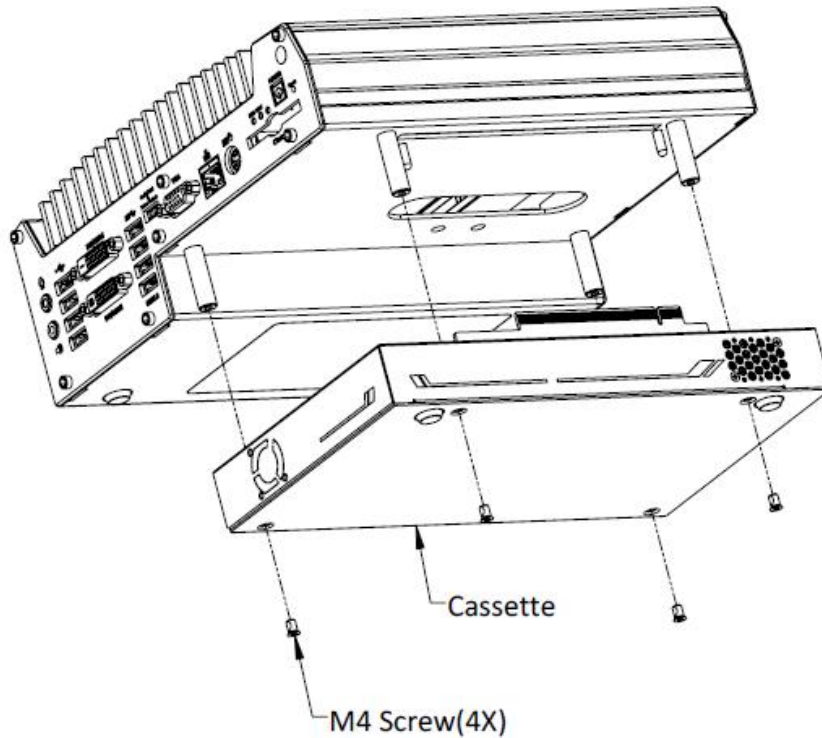
5. Tilt the HDD assembly and insert the wedge of HDD bracket to the bottom cover. Once it's firmly wedged, push it down and fix it using a M3 flat-head screw.



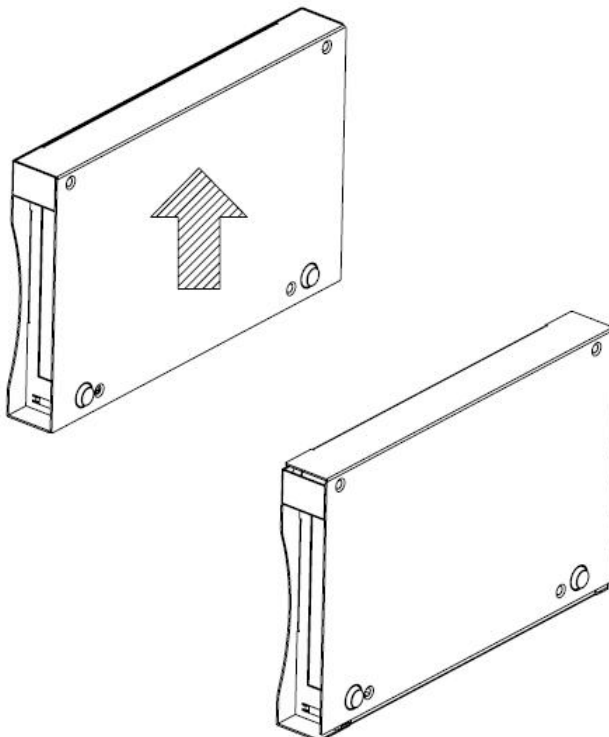
### 3.3 Install an Add-on Card into Cassette

The expansion Cassette is used to accommodate a PCI Express or PCI add-on card. To install an add-on card into Cassette, please refer to the information listed below.

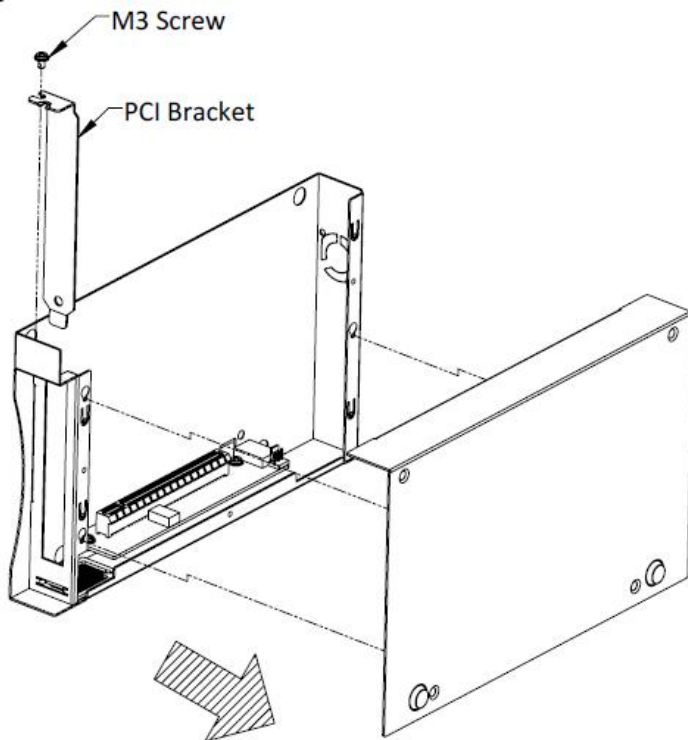
1. Remove Cassette from Nuvis-3304af by loosening four M4 screws.



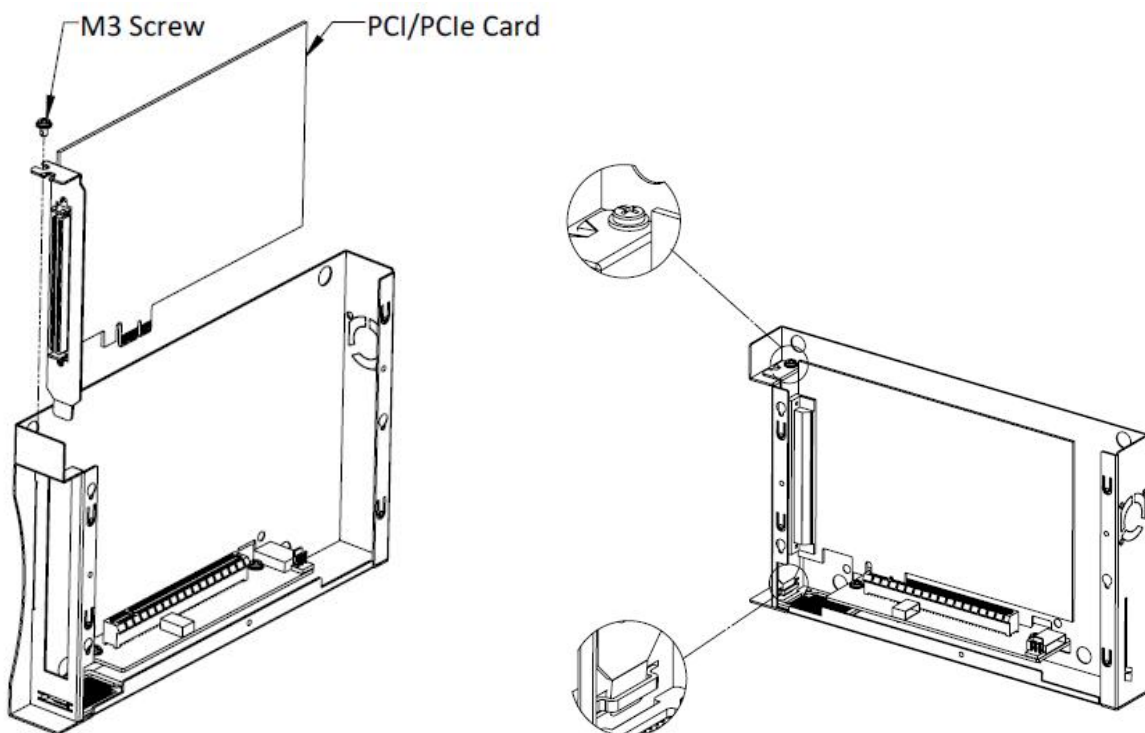
2. Open Cassette by pushing its cover toward arrow pointing direction.



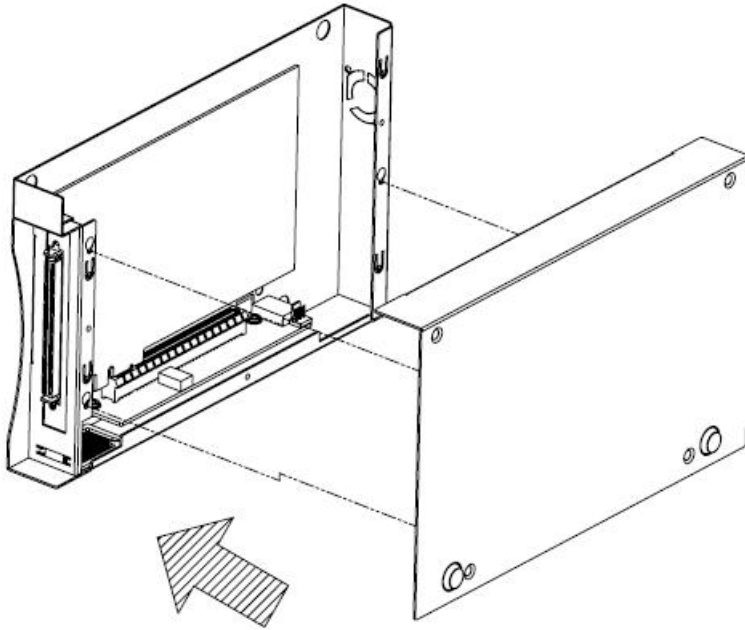
3. Pull the cover of Cassette toward arrow pointing direction to open it. And then remove the blank faceplate installed in Cassette by loosening a M3 screw.



4. Install a PCI/PCIe add-on card into the PCI/PCIe connector. Note that the tail of faceplate of the add-on card must be inserted into the mortise. Tighten the add-on card using a M3 screw.



5. Recover Cassette and assemble it to Nuvis-3304af. Fix Cassette with four M4 screws.



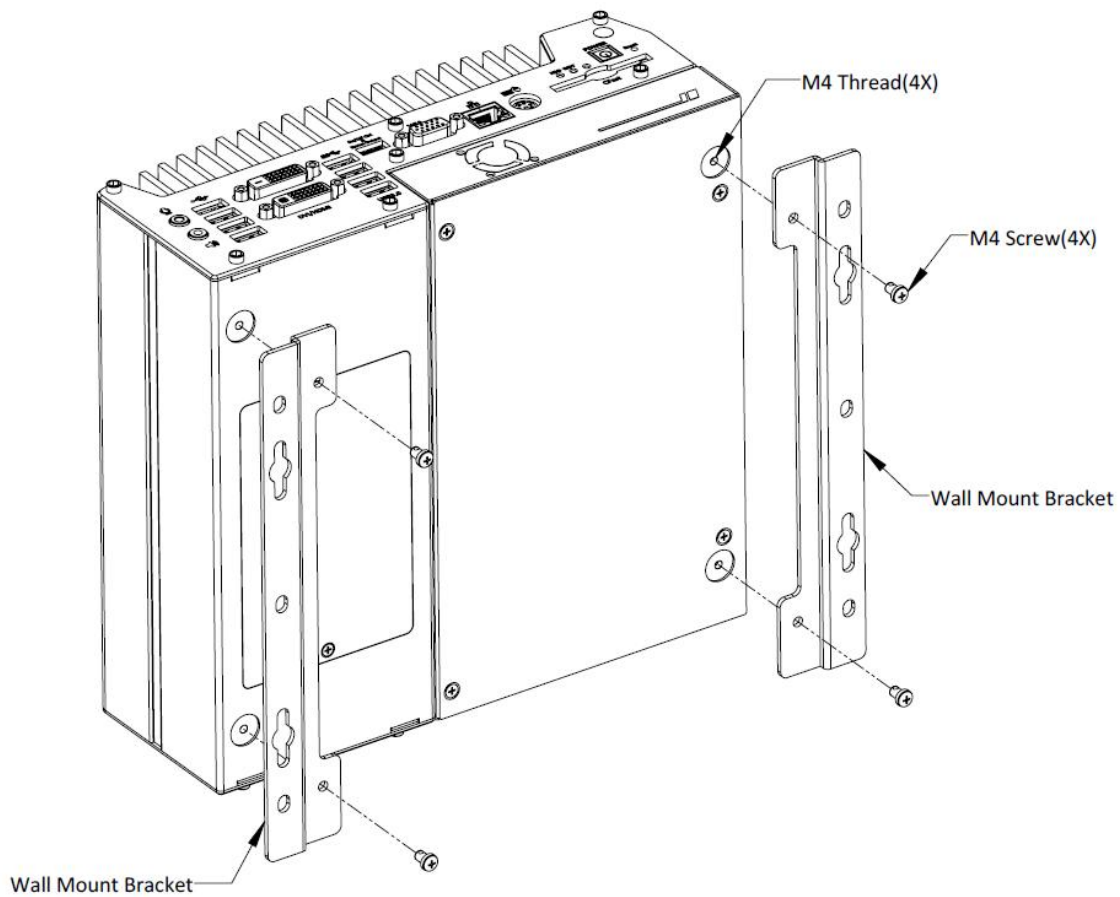
## 3.4 Mount your Nuvis-3304af

Nuvis-3304af provides versatile ways of mounting. You can use wall-mounting brackets shipped with Nuvis-3304af to mount it on the wall. Neousys also offers optional bracket and DIN rail clip to mount Nuvis-3304af on a DIN rail. To mount your Nuvis-3304af, please refer to the information listed below.

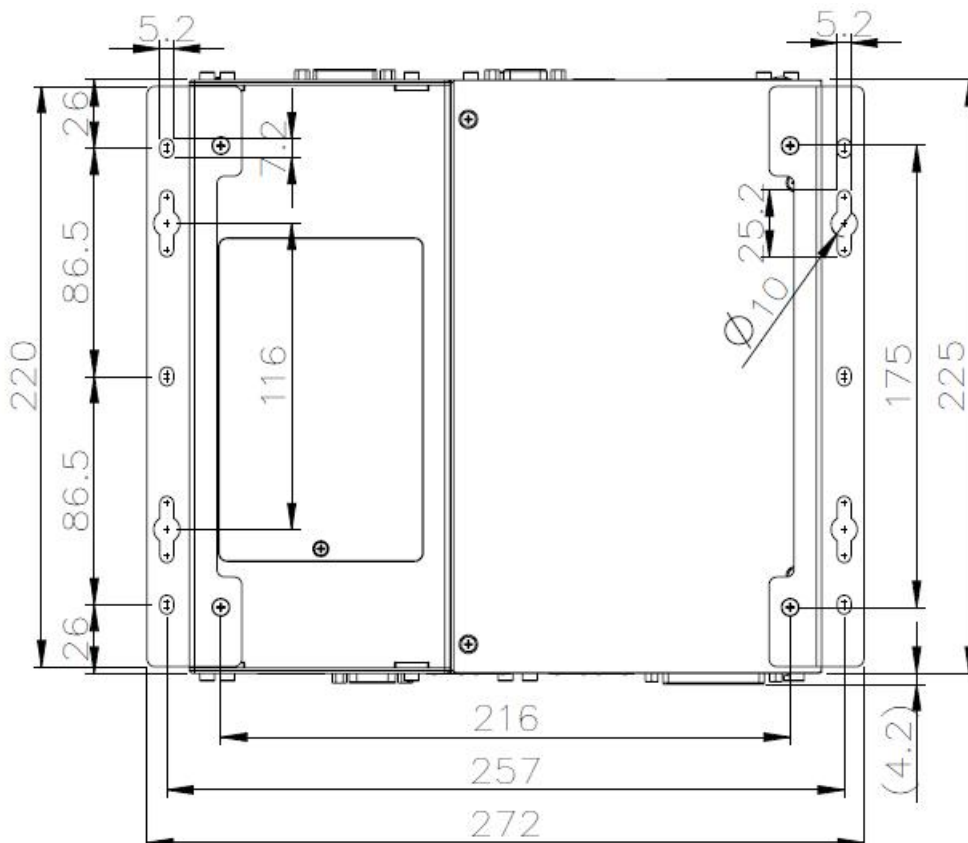
### Mount your Nuvis-3304af on the Wall

Nuvis-3304af is shipped with wall-mount brackets as the standard mounting option. You can mount your Nuvis-3304af on the wall by following the steps listed below.

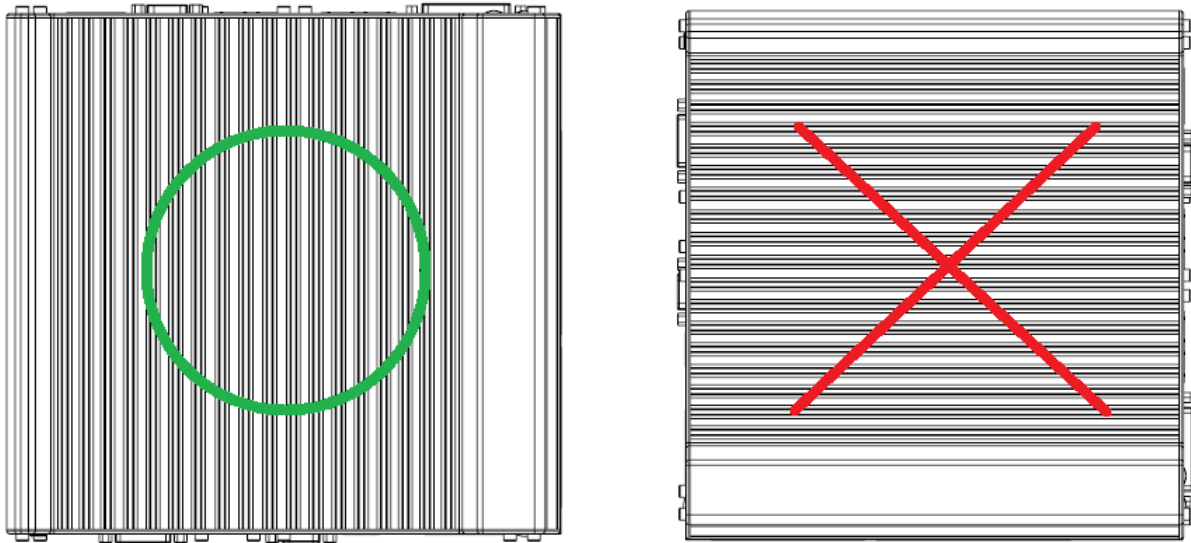
1. Get two wall-mounting brackets and four M4 screws from the accessory box. Fix the mounting brackets to Nuvis-3304af using M4 screws.



2. Place Nuvis-3304af on a flat surface and fix it with screws. You can also take advantage of the keyhole-shaped holes of mounting brackets to suspend Nuvis-3304af on the Wall.

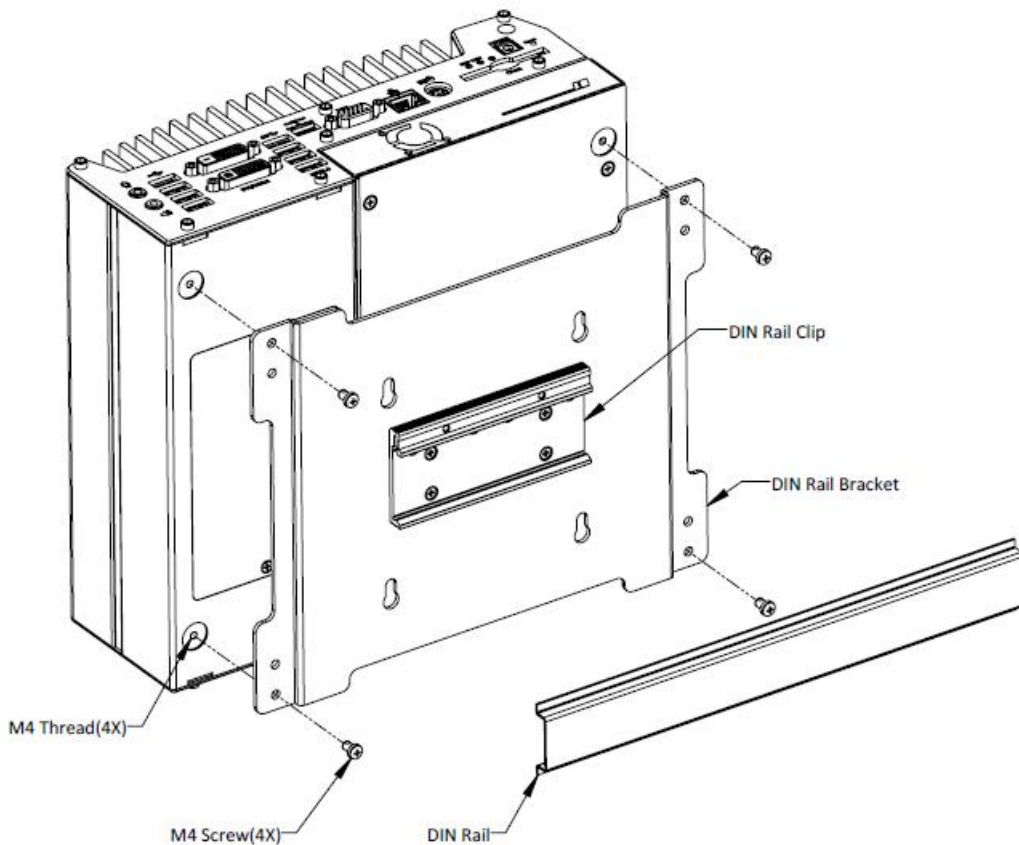


3. For best efficiency of heat dissipation, please mount Nuvis-3304af in a right direction.



## Mount your Nuvis-3304af on the DIN Rail

Neusys also provides the option of the DIN-rail mounting kit. The kit includes a bracket and a DIN-rail mounting clip. You should fix the clip to the bracket using four M4 flat-head screws first, and then fix the bracket assembly to Nuvis-3304af with another four M4 screws. This option can be useful if you want to deploy Nuvis-3304af inside an equipment cabinet where DIN rail is available.



## 3.5 Connect DC power to you Nuvis-3304af

There are two types of power connectors available for Nuvis-3304af. A 4-pin mini-DIN power connector, which is used with an AC/DC adapter, and a 3-pin pluggable terminal block, which is used for direct DC input where DC power is available.

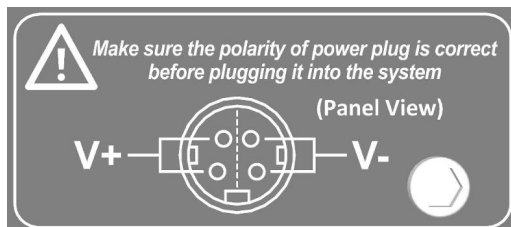
### Caution

1. Please make sure the voltage of DC power is correct before you connect it to Nuvis-3304af. Supplying a voltage over 25V will damage the system.
2. You should use either 4-pin power connector or 3-pin pluggable terminal block for DC power input. **DO NOT** supply power to both connectors at the same time.

### 3.5.1 Connect DC Power via 4-pin Power Connector

The 4-pin mini-DIN power connector provides a convenient way for power input especially in an indoor environment where AC/DC power adapter is usually applied. Neousys provides a 160W AC/DC adapter for powering both Nuvis-3304af system and external PoE devices. To connect DC power via 4-pin mini-DIN power connector, please follow the instructions below.

1. Find one AC-DC adapter with the matched 4-pin mini-DIN power connector. Note that the polarity of the 4-pin mini-DIN power connector must be correct.



2. Make sure the AC-DC adapter is off before you connect the power plug to your Nuvis-3304af.
3. Plug the power plug of adapter into the 4-pin mini-DIN power connector and push it to the end until plug is firmly latched.





## 3.5.2 Connect DC Power via 3-pin Pluggable Terminal Block

For the field usage where DC power is available, the 3-pin pluggable terminal block of Nuvis-3304af provides the reliable way for directly wiring the DC power. To connect DC power via the 3-pin pluggable terminal block, please follow the steps listed below.

1. Make sure the external DC power supply is power off or disconnected before wiring.
2. Get the 3-pin pluggable terminal block from the accessory box. The terminal block fits the wires with a gauge of 12~24 AWG.
3. Carefully identify the positive and negative contacts of your DC power supply and pluggable terminal block. The polarities between DC power supply and terminal block must be positive (+) to positive (+) and ground (GND) to ground (GND).



4. Insert the wires to correct contacts of pluggable terminal block and tighten clamping screws using a Philips screwdriver.



5. Firmly plug the terminal block into a receptacle on the rear panel, and tighten the captive screws using a Slotted screwdriver.

### Caution

1. Nuvis-3304af accepts 8~25 VDC when using terminal block for DC input. Please make sure the voltage of DC power is correct before you connect it to Nuvis-3304af. Supplying a voltage over 25V will damage the system.

## 3.6 Power on your Nuvis-3304af

For better flexibility of operation, Nuvis-3304af provides three alternatives to power on your Nuvis-3304af. You can turn on your Nuvis-3304af by pressing the power button, using an external non-latched on/off switch, or by sending a special LAN packet. In this section, we illustrate these ways to power on your Nuvis-3304af.

### 3.6.1 Power on Nuvis-3304af Using the Power Button

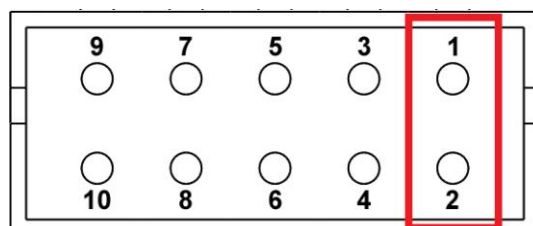
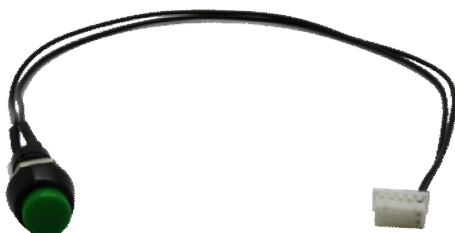
This is the simplest way to turn on your Nuvis-3304af. The power button on the front panel is a non-latched switch and behaves the ATX-mode on/off control. As DC power is connected, push the power button will turn on the system as well as the PWR LED indicator. Push the button when system is on will turn off the system. If your operating system supports ATX power mode (i.e. Microsoft Windows or Linux), push the power button causes a pre-defined system behavior, such as shutdown or hibernation.



### 3.6.2 Power on Nuvis-3304af Using an External Non-latched Switch

For an application which places Nuvis-3304af inside a cabinet, it's useful to control the on/off of the system using an external switch. Nuvis-3304af provides a 2x5, 2.0mm pitch wafer connector (for detail, please refer to section 2.2.8) for connecting a non-latched switch and behaves ATX-mode power on/off control. The external non-latched switch acts exactly the same as the power button on the front panel. To power on Nuvis-3304af using an external non-latched switch (ATX-mode), please follow the steps listed below.

1. Prepare a non-latched switch with a 2x5, 2.0mm pitch wafer terminal. Note that the switch must be connected to pin#1 and pin#2 (polarity is negligible).



2. Connect the wafer terminal to the wafer connector on the front panel of Nuvis-3304af.



3. Push the non-latched switch will turn on the system (the PWR LED indicator on the front panel is on at the same time). Push the non-latched switch when system is on will turn off the system. If your operating system supports ATX power mode (i.e. Microsoft Windows or Linux), push the power button causes a pre-defined system behavior, such as shutdown or hibernation.

### 3.6.3 Power on Nuvis-3304af Using Wake-on-LAN Function

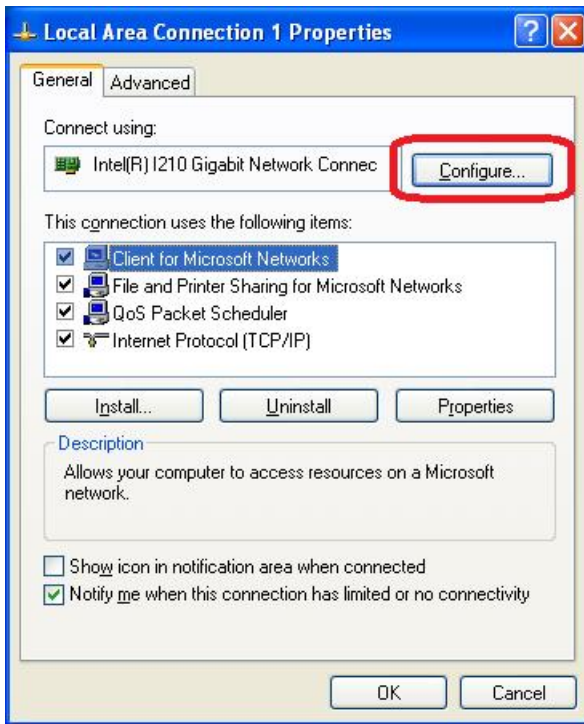
Wake-on-LAN (WOL) is a mechanism to wake up a computer system from a S5 (system off with standby power) state via issuing a magic packet. Nuvis-3304af implements the Wake-on-LAN function for its first GbE port.



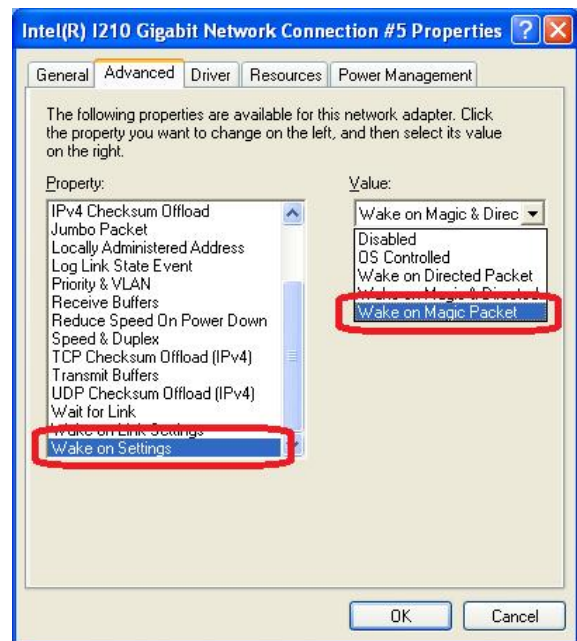
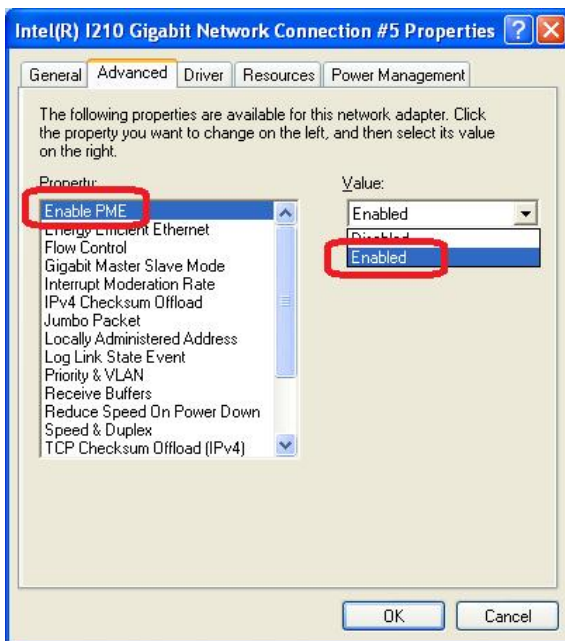
To enable WOL function and power on you Nuvis-3304af, please follow the steps listed below.

#### For Windows XP

1. When Nuvis-3304af boots up, press F2 to enter BIOS setup utility.
2. Enter the **[Power]** menu. And configure the **[Wake On LAN]** option as **[Enabled]**. This setting enables the Wake-on-LAN function for Nuvis-3304af. Please refer to section 4.1.4 for the instruction of configuring this BIOS option.
3. In Windows XP system, identify the Local Area Connection of corresponding Intel® I210 Gigabit Controller and click the **Configure** button.



4. Click the **Advanced** tag, and configure the following two options accordingly.



- **Enable PME**

This option enables Intel® I210 to generate a power management event and signal the Nuvis-3304af to wake up from S5 state.

- **Wake on Magic Packet**

Nuvis-3304af can wake from S5 state when receiving a magic packet. The magic packet is a broadcast frame containing anywhere within its payload 6 bytes of all 255 (FF FF FF FF FF FF in hexadecimal), followed by sixteen repetitions of the target computer's 48-bit MAC address.



For example, NIC's 48-bit MAC Address is 78h D0h 04h 0Ah 0Bh 0Ch

```

DESTINATION SOURCE MISC
FF FF FF FF FF FF
78 D0 04 0A 0B 0C 78 D0 04 0A 0B 0C
78 D0 04 0A 0B 0C 78 D0 04 0A 0B 0C
78 D0 04 0A 0B 0C 78 D0 04 0A 0B 0C
78 D0 04 0A 0B 0C 78 D0 04 0A 0B 0C
78 D0 04 0A 0B 0C 78 D0 04 0A 0B 0C
78 D0 04 0A 0B 0C 78 D0 04 0A 0B 0C
78 D0 04 0A 0B 0C 78 D0 04 0A 0B 0C
78 D0 04 0A 0B 0C 78 D0 04 0A 0B 0C
MISC    CRC

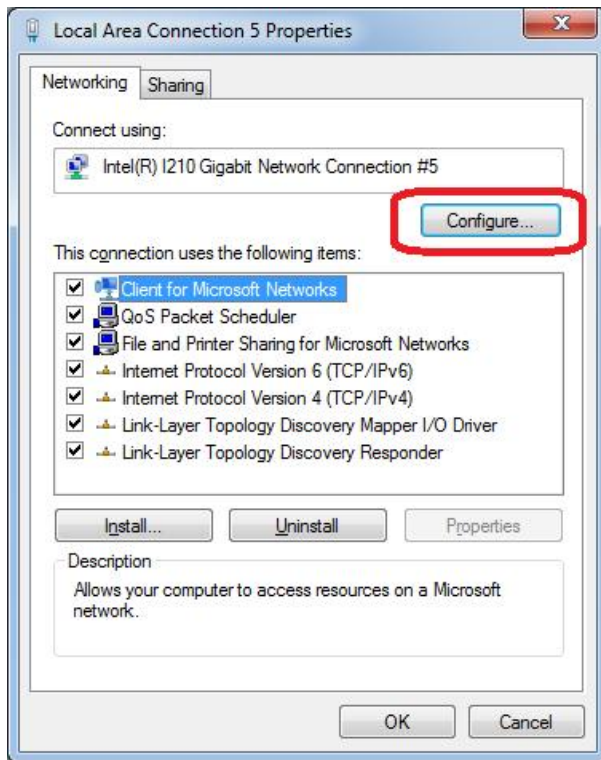
```

There are some free tools available on Internet that can be used to send a magic packet. Please refer to the following link to understand more about Magic Packet.  
<http://en.wikipedia.org/wiki/Wake-on-LAN>

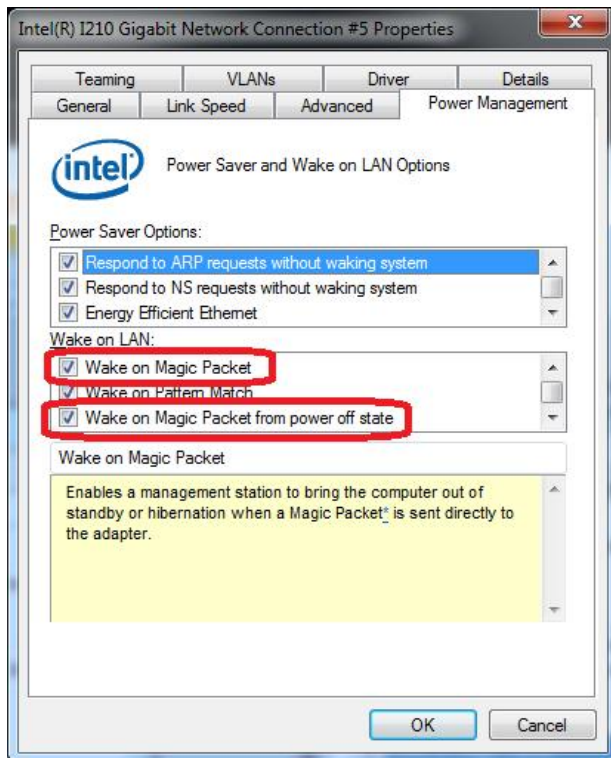
### For Windows 7

1. When Nuvis-3304af boots up, press F2 to enter BIOS setup utility.
2. Enter the **[Power]** menu. And configure the **[Wake On LAN]** option as **[Enabled]**. This setting enables the Wake-on-LAN function for Nuvis-3304af. Please refer to section 4.1.4 for the instruction of configuring this BIOS option.
3. In Windows 7 system, identify the Local Area Connection of corresponding Intel® I210 Gigabit Controller and click the **Configure** button.





4. Click the **Power Management** tag, and check the following two options accordingly



# Chapter 4 BIOS and Driver

## 4.1 BIOS Settings

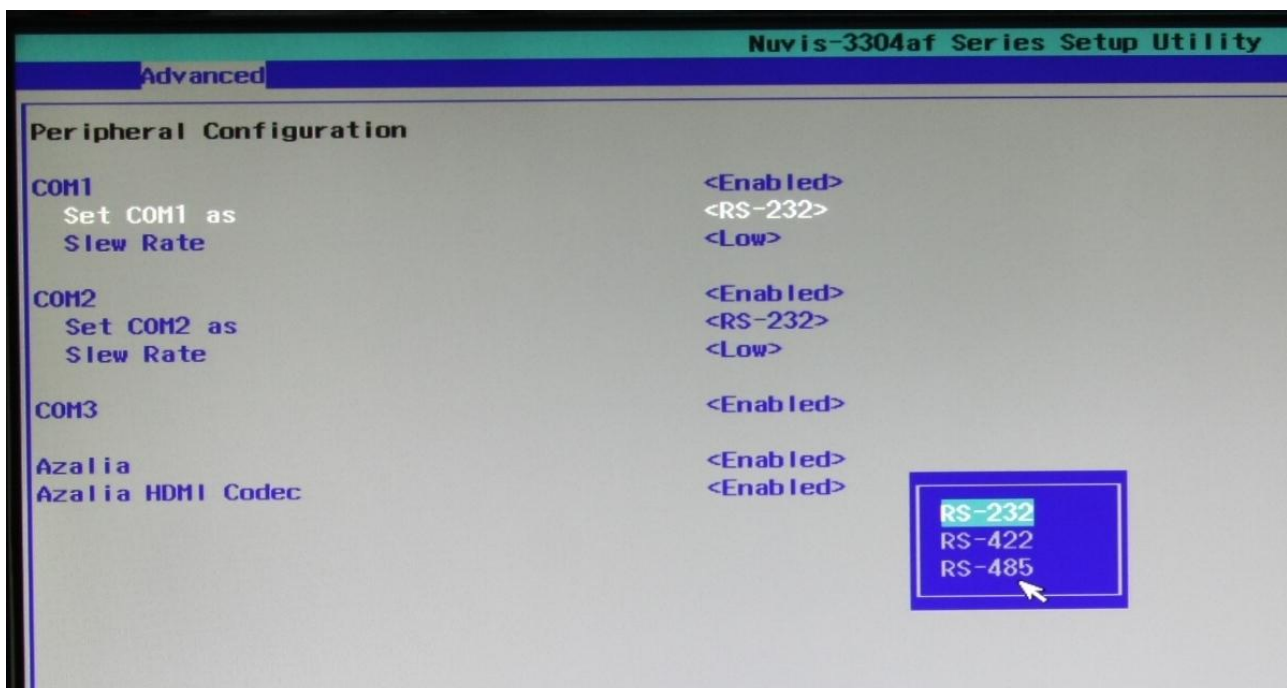
Nuvis-3304af is shipped with factory-default BIOS settings cautiously programmed for best performance and compatibility. In this section, we'll illustrate some of BIOS settings you may need to modify. Please always make sure you understand the effect of change before you proceed with any modification.

### 4.1.1 COM1 & COM2 Operating Mode

COM1 and COM2 of Nuvis-3304af support RS-232 (full-duplex), RS-422 (full-duplex) and RS-485 (half-duplex) mode. You can set the COM1/COM2 operating mode via BIOS settings. Another option in BIOS called "*Slew Rate*" defines how sharp the rising/falling edge is for the output signal of COM1/COM2. For long-distance RS-422/485 transmission, you may set the "*Slew Rate*" option as "High" to improve signal quality.

To set COM1/COM2 operating mode:

1. When Nuvis-3304af boots up, press F2 to enter BIOS setup utility.
2. Go to **[Advanced]** → **[Peripheral Configuration]**.
3. Set the **[Set COM1 as]** to a proper mode for COM1 of your Nuvis-3304af.
4. Set the **[Set COM2 as]** to a proper mode for COM2 of your Nuvis-3304af.



## 4.1.2 SATA Controller Mode

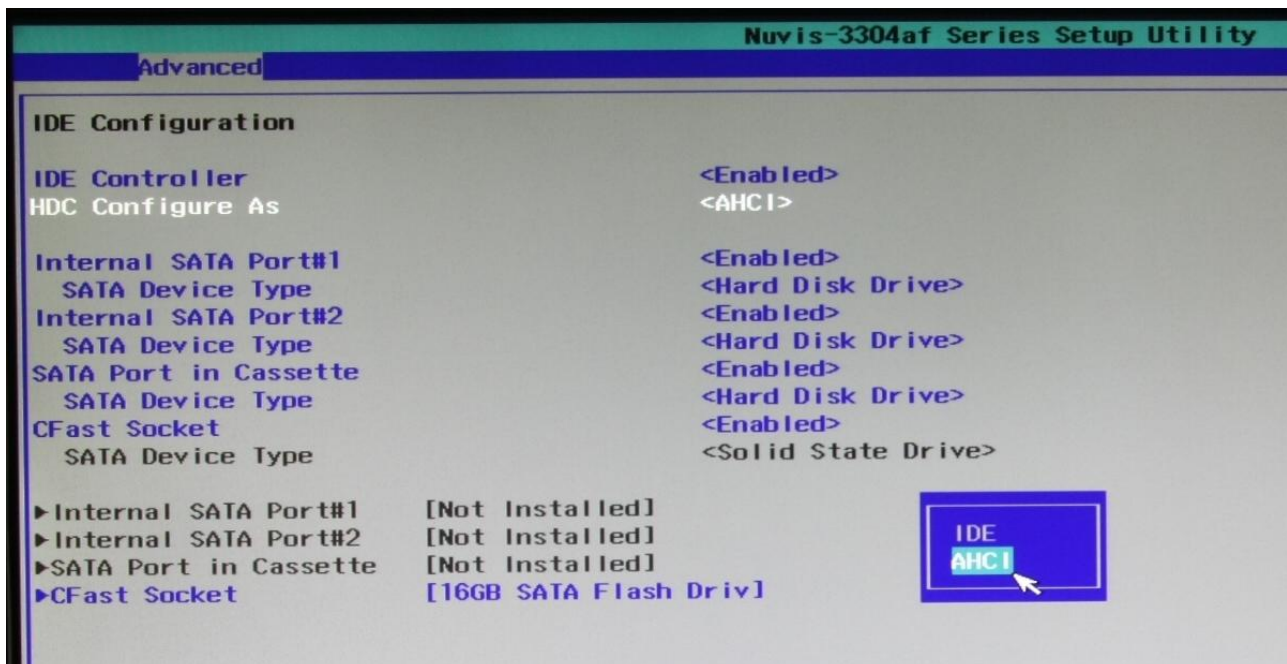
The SATA controller of Nuvis-3304af supports two modes of operations, **IDE** and **AHCI** mode. **IDE** mode configures SATA controller to access SATA interface in legacy IDE mode, and is compatible with most storage devices. **AHCI** mode, which exposes SATA's advanced capabilities such as hot swapping and native command queuing, is supported in several later version of operating systems.

Our suggestion of how to set SATA controller mode is

- If you're using Windows XP, Linux kernel earlier than 2.6.19, you should select **IDE** mode.
- If you're using Windows Vista, Windows 7, Windows 8, or Linux kernel from 2.6.19 onward, you can select **AHCI** mode for better performance.

To set SATA controller mode:

1. When Nuvis-3304af boots up, press F2 to enter BIOS setup utility.
2. Go to **[Advanced]** → **[SATA Configuration]**.
3. Set the **[HDC Configure as]** to a proper mode for your Nuvis-3304af.





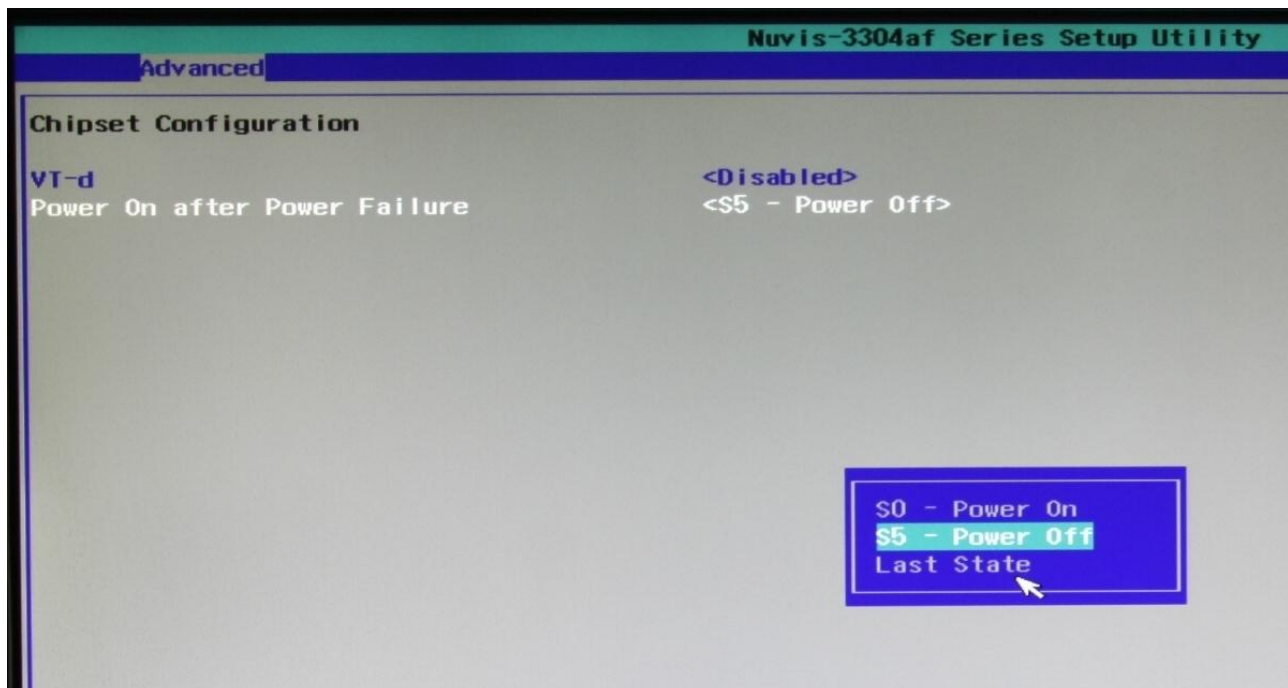
### 4.1.3 Power On after Power Failure Option

This option defines the behavior of Nuvis-3304af when DC power is supplied.

Value	Description
S0 – Power On	System is powered on when DC power is supplied.
S5 – Power Off	System is kept in off state when DC power is supplied.
Last State	The on/off state of the system is determined according to the last state when DC power is disconnected. For example, if system is still on but DC power is unplugged, the system is powered on next time when DC power is supplied.

To set “Power On after Power Failure” option:

1. When Nuvis-3304af boots up, press F2 to enter BIOS setup utility.
2. Go to **[Advanced]** → **[Chipset Configuration]**.
3. Set the **[Power On after Power Failure]** to a proper value for your Nuvis-3304af.



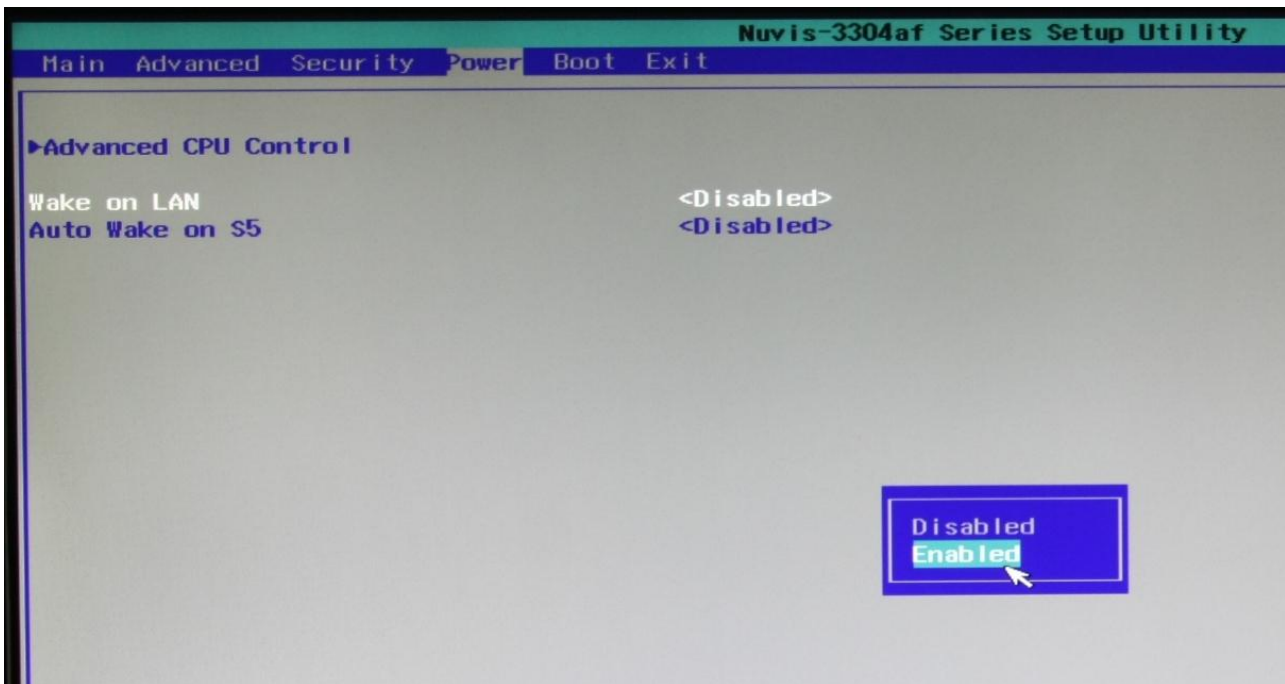
### 4.1.4 Wake-on-LAN Option

Wake-on-LAN (WOL) is a mechanism which allows you to turn on your Nuvis-3304af via Ethernet connection. To utilize Wake-on-LAN function, you have to enable this option first in BIOS settings. Please refer to section 3.6.3 for instructions of using WOL function.

To enable/disable “Wake on LAN” option:

1. When Nuvis-3304af boots up, press F2 to enter BIOS setup utility.

2. Go to **[Power]**.
3. Enable/disable the **[Wake on LAN]** option according to your application.

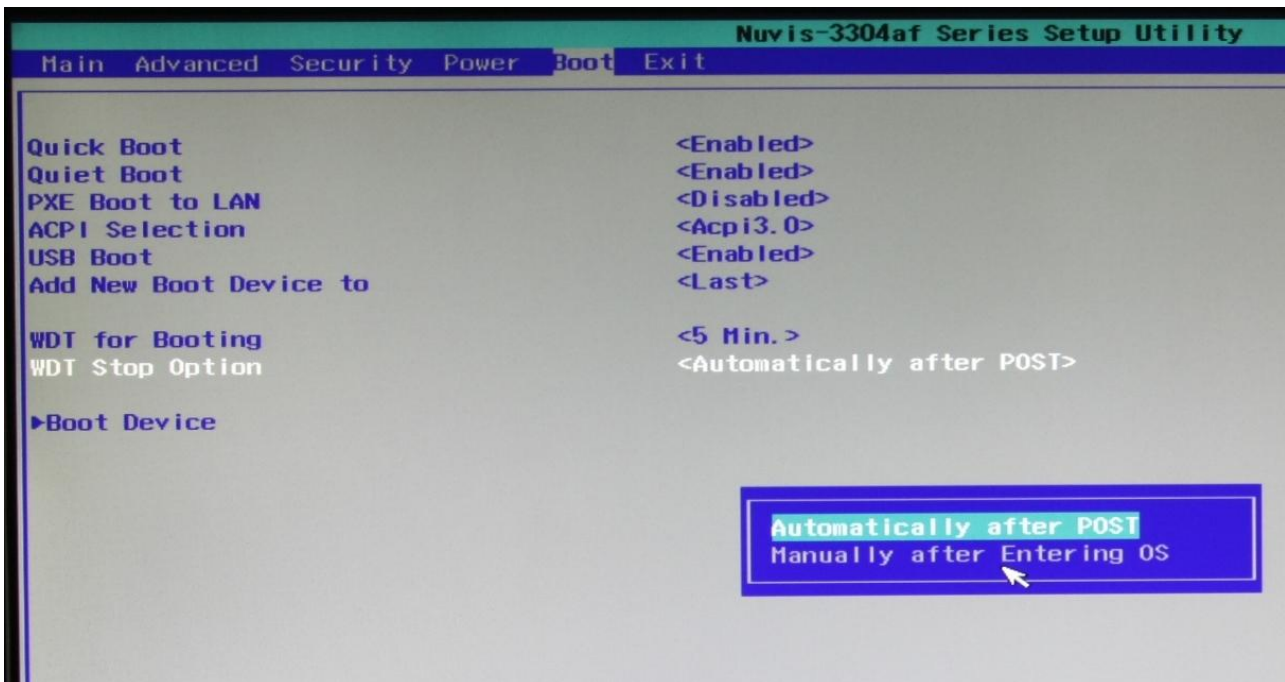
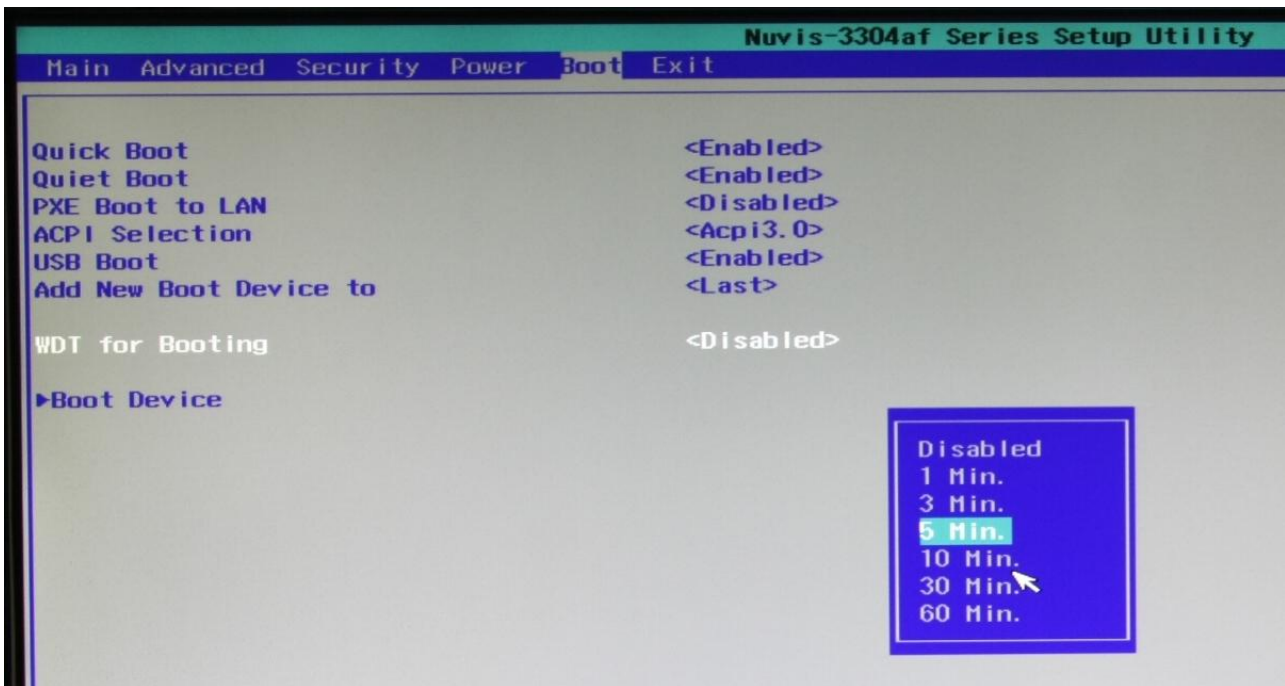


### 4.1.5 Watchdog Timer for Booting

The BIOS of Nuvis-3304af has a useful feature which allows users to use the watchdog timer to secure the booting process. You can specify the timeout value for watchdog timer. Once the watchdog timer expires, the BIOS issues a reset command to initiate another booting process. You can also set the behavior of how to stop the watchdog timer. There are two options in BIOS menu, “*Automatically after POST*” and “*Manually after Entering OS*”. When “*Automatically after POST*” is selected, the BIOS automatically stop the watchdog timer after POST (Power-On Self Test) OK. When “*Manually after Entering OS*” is selected, it’s user’s responsibility to stop the watchdog timer when entering OS. This guarantees the system can always boot to OS, otherwise another booting process will be initiated. For information about programming watchdog timer, please refer to **Appendix A Using Watchdog Timer & Isolated DIO**.

To set the watchdog timer for boot in BIOS:

1. When Nuvis-3304af boots up, press F2 to enter BIOS setup utility.
2. Go to **[Boot]** menu.
3. Disable or select timeout value for **[WDT for Booting]** option.
4. Once you give a timeout value, the **[WDT Stop Option]** option appears. You can select “*Automatically after POST*” or “*Manually after Entering OS*”.



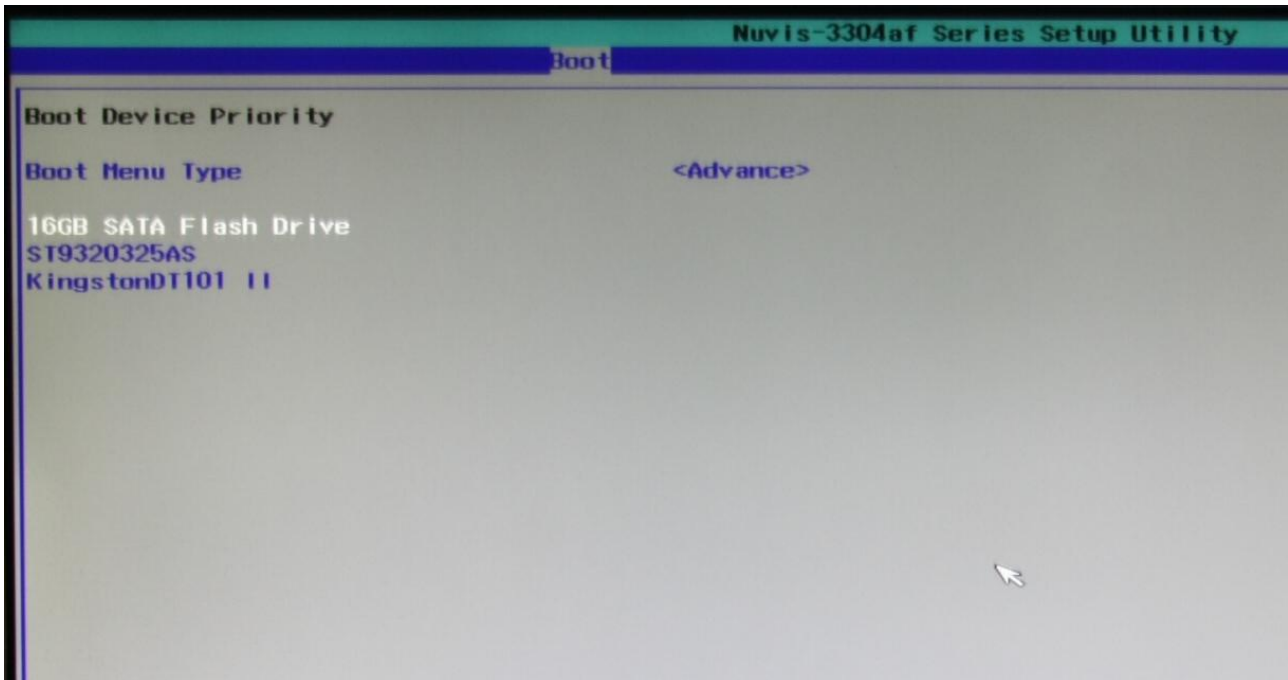
## 4.1.6 Select a Boot Device

When you have multiple bootable devices connected to your Nuvis-3304af (i.e. HDD, mSATA, USB flash disk, USB DVD-drive), you may need to select one of them as the boot device. There are two ways to select the device. You can either, press F12 when system boots up to go to Boot Manager and then select one of the devices, or select the boot device in BIOS settings.

To select a boot device in BIOS:



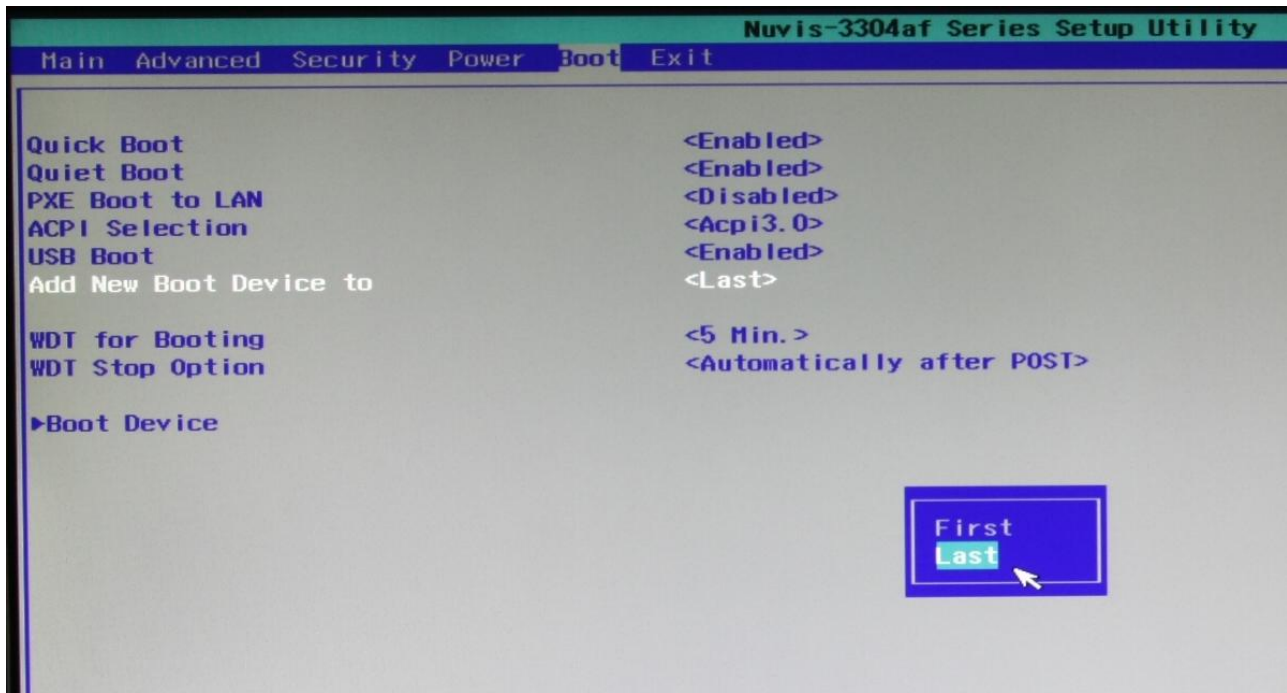
1. When Nuvis-3304af boots up, press F2 to enter BIOS setup utility.
2. Go to **[Boot] → [Boot Device]**.
3. The **[Boot Menu Type]** option decides whether to list all bootable devices connected to your Nuvis-3304af according to device (**Advanced** mode) or device category (**Normal** mode). You can use F5/F6 or +/- to change the boot order of devices or device categories.



### 4.1.7 Position New Boot Device

When you plug a new boot device (e.g. USB flash disk, CFAST and etc.), this option allow you whether to place the new boot device in the first place or the last place in the boot order queue.

1. When Nuvis-3304af boots up, press F2 to enter BIOS setup utility.
2. Go to **[Boot]** menu.
3. The **[Add New Boot Device]** option decides whether to place the new boot device to the **First** place or the or the **Last** place in the boot order queue.



## 4.2 Operating System Support

Nuvis-3304af supports most operating system developed for Intel® x86 architecture. The following list contains the operating systems which have been tested in Neousys Technology Inc.

- Microsoft Window XP 32-bit
- Microsoft Window 7 32-bit
- Microsoft Window 7 64-bit
- Ubuntu (version later than 11.04 Desktop 64bit)\*
- Ubuntu (version later than 10.10 Desktop 32bit)\*
- Debian (version later than 6.0 64bit)\*
- Fedora 14 i386 32bit\*
- CentOS 6.4 32bit

*\*For Linux system, user may need to manually compile and install the driver for Intel I210 GbE controller if the driver is not embedded in kernel. You can visit Intel website for further information.*

Neousys will keep this list updated as we continuously test other operating systems with Nuvis-3304af. Please contact us for the latest OS support list.



## 4.3 Driver Installation

Neosys Technology Inc. provides a very convenient utility in “Drivers & Utilities DVD” to allow the “One-Click” driver installation. This utility automatically detects your Windows operating system and installs all necessary drivers to your Nuvis-3304af with just one mouse click.

### 4.3.1 Install All Drivers Using “One-Click” Driver Installation

1. Insert the “Drivers & Utilities DVD” into a USB DVD-drive attached to your Nuvis-3304af. A setup utility launches and the following dialog appears.



2. Click on the “**Automatic Driver Installation**”. The setup utility will automatically detect your Windows operating system and install all necessary drivers. According to different versions of Windows, the installation process takes about 6~8 minutes. Once driver installation is done, the setup utility reboots your Windows and your system works normally afterward.

## 4.3.2 Install Drivers Manually

You can also manually install each driver for Nuvis-3304af. Please refer to the following information about installing drivers for different operating system.

### Windows XP SP3

The recommended driver installation sequence is

1. Chipset driver (x:\Driver\_Pool\Chipset\_7\_Series\ALL\infinst\_autol.exe)
2. .NET Framework 3.5 (x:\Driver\_Pool\DotNET35\XP\dotnetfx35.exe)
3. Graphics driver (x:\Driver\_Pool\Graphics\_3rd\_i7\XP\Setup.exe)
4. Audio driver (x:\Driver\_Pool\Audio\_ALC262\XP\Setup.exe)
5. LAN driver (x:\Driver\_Pool\GbE\_I210\XP\I210\_XP\_Install.exe)
6. ME driver (x:\Driver\_Pool\ME\_7\_Series\ALL\Setup.exe)

### Windows XP SP2

The recommended driver installation sequence is

1. Chipset driver (x:\Driver\_Pool\Chipset\_7\_Series\ALL\infinst\_autol.exe)
2. Windows Installer 3.1  
(x:\Driver\_Pool\DotNET35\XP\WindowsInstaller-KB893803-v2-x86.exe)
3. .NET Framework 3.5 (x:\Driver\_Pool\DotNET35\XP\dotnetfx35.exe)
4. Audio patch for HD audio  
(x:\Driver\_Pool\Audio\_ALC262\XP\MSHDQFE\Win2K\_XP\us\kb888111xpsp2.exe)
5. Graphics driver (x:\Driver\_Pool\Graphics\_3rd\_i7\XP\Setup.exe)
6. Audio driver (x:\Driver\_Pool\Audio\_ALC262\XP\Setup.exe)
7. LAN driver (x:\Driver\_Pool\GbE\_I210\XP\I210\_XP\_Install.exe)
8. ME driver (x:\Driver\_Pool\ME\_7\_Series\ALL\Setup.exe)

### Windows 7 32-bit

The recommended driver installation sequence is

1. Chipset driver (x:\Driver\_Pool\Chipset\_7\_Series\ALL\infinst\_autol.exe)
2. Graphics driver (x:\Driver\_Pool\Graphics\_3rd\_i7\Win7\_32\Setup.exe)
3. Audio driver (x:\Driver\_Pool\Audio\_ALC262\Win7\_ALL\Setup.exe)
4. LAN driver  
(x:\Driver\_Pool\GbE\_I210\Win7\_32\APPS\PROSETDX\Win32\DxSetup.exe)
5. USB 3.0 driver (x:\Driver\_Pool\USB3\_7\_Series\Win7\_ALL\Setup.exe)
6. ME driver (x:\Driver\_Pool\ME\_7\_Series\ALL\Setup.exe)





### Windows 7 64-bit

The recommended driver installation sequence is

1. Chipset driver (x:\Driver\_Pool\Chipset\_7\_Series\ALL\infinst\_autol.exe)
2. Graphics driver (x:\Driver\_Pool\Graphics\_3rd\_i7\Win7\_64\Setup.exe)
3. Audio driver (x:\Driver\_Pool\Audio\_ALC262\Win7\_ALL\Setup.exe)
4. LAN driver  
(x:\Driver\_Pool\GbE\_I210\Win7\_64\APPS\PROSETDX\Winx64\DxSetup.exe)
5. USB 3.0 driver (x:\Driver\_Pool\USB3\_7\_Series\Win7\_ALL\Setup.exe)
6. ME driver (x:\Driver\_Pool\ME\_7\_Series\ALL\Setup.exe)

### 4.3.3 Install Driver for DTIO and Per-port PoE On/Off Control

Neosys provides a driver package which contain function APIs for programming DTIO and per-port PoE power on/off control function. You should install the driver package (WDT\_DIO\_Setup.exe) in prior to use these functions. Please note that you must install WDT\_DIO\_Setup\_v2.0.3 or later revision for Nuvis-3304af.

#### Windows XP SP3

Please execute the driver setup program in the following directory.

x:\Driver\_Pool\WDT\_DIO\XP\_Win7\_32\WDT\_DIO\_Setup\_v2.0.3.exe

#### Windows 7 32-bit

Please execute the driver setup program in the following directory.

x:\Driver\_Pool\WDT\_DIO\XP\_Win7\_32\WDT\_DIO\_Setup\_v2.0.3.exe

#### Windows 7 64-bit

Please execute the driver setup program in the following directory.

x:\Driver\_Pool\WDT\_DIO\Win7\_64\WDT\_DIO\_Setup\_v2.0.3(x64).exe



# Appendix A Using Watchdog Timer & DTIO

Neosys Nuvis-3304af series provide a watchdog (WDT) timer function to ensure a more reliable system operation. The WDT is a hardware mechanism to reset the system if the watchdog timer is expired. Users can start the WDT and keeping resetting the timer to make sure the system or program is running. Otherwise, the system shall be reset.

Except for hardware watchdog function, Nuvis-3304af features the DTIO technology to provide accurate trigger/strobe control for machine vision applications. It provides 8-CH isolated DI and 8-CH isolated DO via a DSub-25 connector. DTIO supports versatile modes such as polling I/O with Change-of-State interrupt, deterministic trigger I/O and deterministic trigger fan-out. Users can program the DTIO function via Neosys' WDT & DIO driver package.

In this document, we'll illustrate how to use the function library provided by Neosys to program the WDT and DTIO functions. Currently, WDT & DIO driver library supports the following operating systems:

- Windows XP
- Windows Vista (32-bit)
- Windows 7 (32-bit & 64-bit)

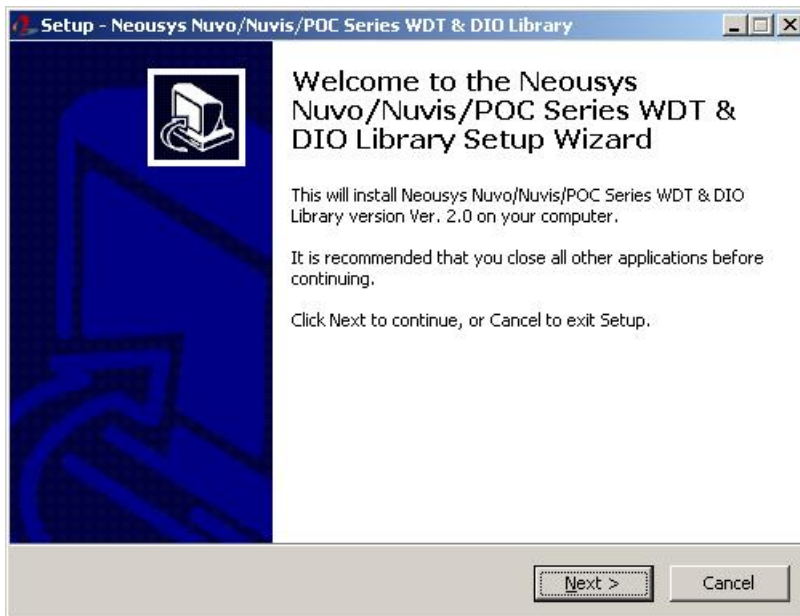
For other OS support, please contact Neosys Technology for further information.



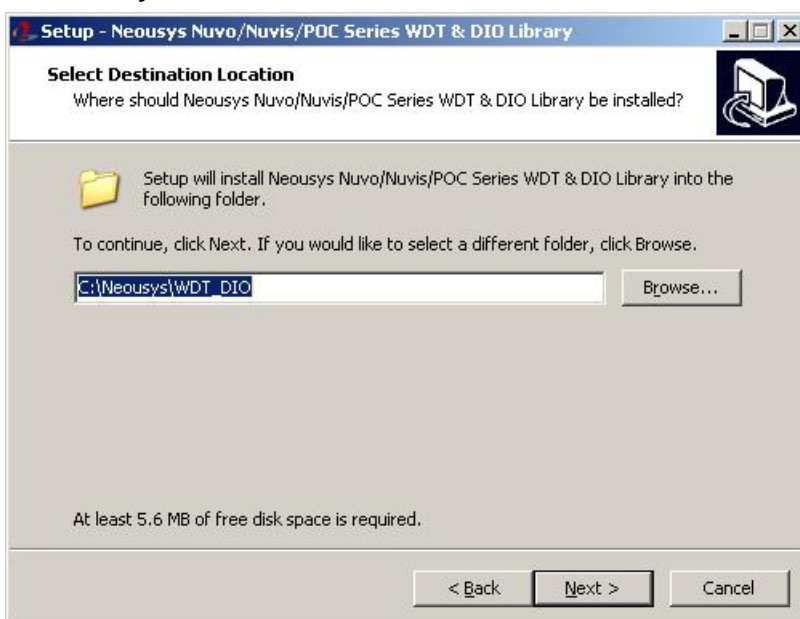
# Install WDT and DIO Library

The WDT\_DIO function library is delivered in the form of a setup package named **WDT\_DIO\_Setup.exe**. In prior to program WDT and DTIO, you should execute the setup program and install the WDT and DIO library. Please note that you must use WDT\_DIO\_Setup\_v2.0.3 or later revision for Nuvis-3304af.

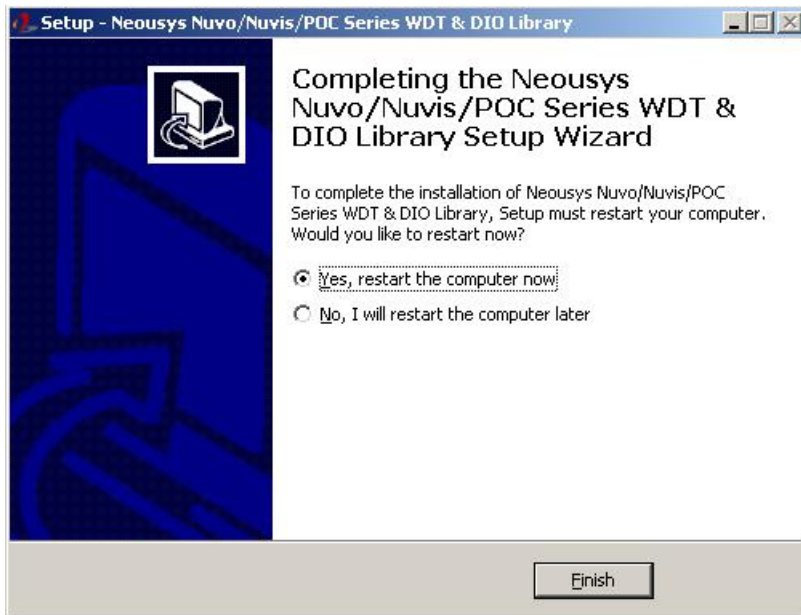
1. Execute **WDT\_DIO\_Setup.exe**. The following dialog appears.



2. Click "Next >" and specify the directory of installing related files. The default directory is `C:\Neosys\WDT_DIO`.



3. Once the installation is finished, a dialog appears to prompt you to reboot the system. The WDT & DIO library will take effect after system rebooting.



4. When you programming your WDT or DIO program, the related files are located in
- Header file:            \Include
  - Lib file:                \Lib
  - Function Reference:    \Manual
  - Sample Code:            \Sample\WDT\_Demo (Watchdog Demo)  
                               \Sample\DIO\_Demo (Polling I/O Demo)  
                               \Sample\COS\_Demo (Change-of-State Interrupt Demo)  
                               \Sample\DTIO\_Demo (DTIO Demo)  
                               \Sample\DTFO\_Demo (Trigger Fan-out Demo)  
                               \Sample\PoE\_Demo (PoE per-port Control Demo)

# Using WDT Function

## WDT Function Reference

### InitWDT

#### Description

Initialize the WDT function. You should always invoke InitWDT() before set or start watchdog timer.

#### Parameter

None

#### Return Value

Returns TRUE if initialization successes, FALSE if initialization failed.

#### Usage

```
BOOL bRet = InitWDT();
```

### SetWDT

#### Description

Set timeout value and unit for watchdog timer. When InitWDT() is invoked, a default timeout value of 255 seconds is assigned.

#### Parameter

*tick*

WORD value (1 ~ 65535) to indicate timeout ticks.

*unit*

BYTE value (0 or 1) to indicate unit of timeout ticks.

0 : unit is minute

1: unit is second

#### Return Value

If value of unit is correct (0 or 1), this function returns TRUE, otherwise FALSE.

#### Usage

```
WORD tick=255;
```

```
BYTE unit=1; //unit is second.
```

```
BOOL bRet = SetWDT(tick, unit); //timeout value is 255 seconds
```



## StartWDT

### Description

Start countdown of WDT. When WDT is started, the WDT LED indicator starts to blink in a frequency of 1Hz. If no ResetWDT() or StopWDT is invoked before WDT is counted to 0, the WDT expires and system resets.

### Parameter

None

### Return Value

If the timeout value is given in correct format, this function returns TRUE, otherwise FALSE.

### Usage

BOOL bRet = StartWDT()

## ResetWDT

### Description

Reset the timeout value to the value given by SetWDT(). If no ResetWDT() or StopWDT is invoked before WDT is counted to 0, the WDT expires and system resets.

### Parameter

None

### Return Value

Always returns TRUE;

### Usage

BOOL bRet = ResetWDT()

## StopWDT

### Description

Stop the countdown of WDT. When WDT is stopped, the WDT LED indicator stops blinking.

### Parameter

None

### Return Value

Always returns TRUE;

### Usage

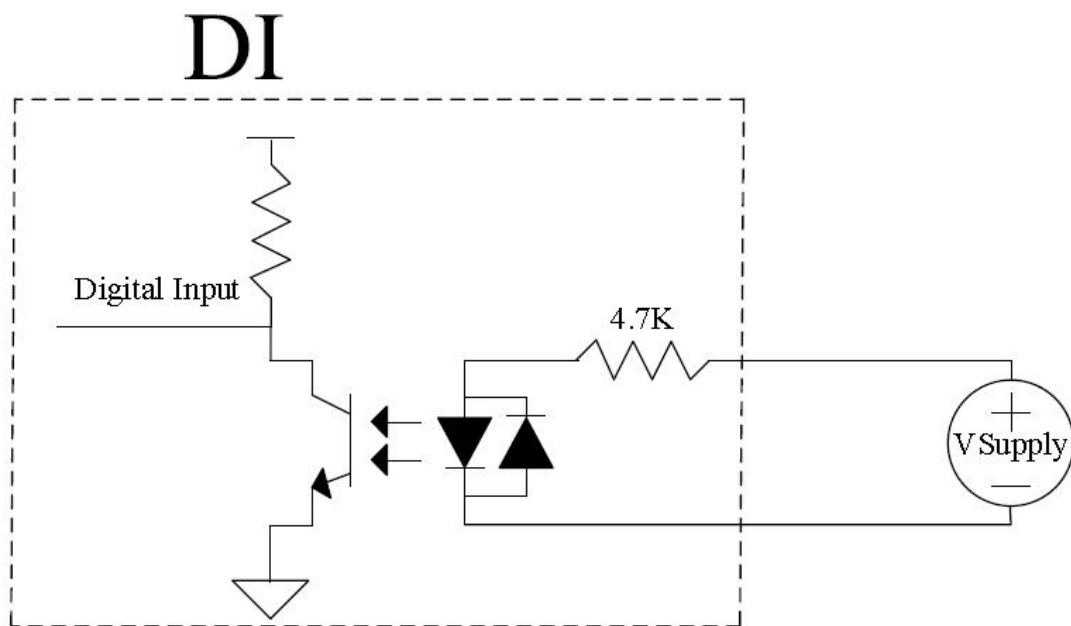
BOOL bRet = StopWDT()



# Using DIO Function

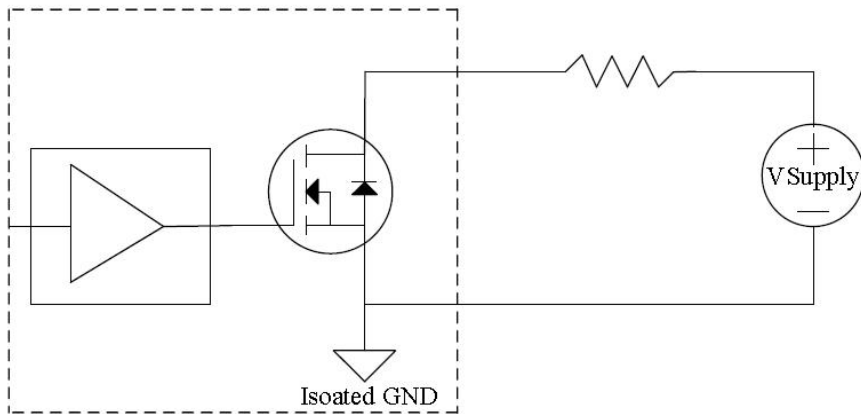
## Wiring for DIO

The digital input function of Nuvis-3304af is implemented using a photo-coupler with a internally series-connected 4.7k $\Omega$  resistor. You need to provide a voltage to specify the logic high/low state. The input voltage for logic high is 5~24V, and the input voltage for logic low is 0~1.5V.



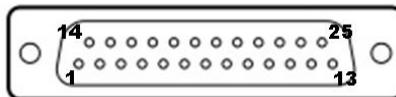
The digital output function of Nuvis-3304af is implemented using Power MOSFET + Analog Device iCoupler® component. The DO channels are configured as NO (normally-open) configuration. When you turn on system, all DO channels have a deterministic state of logic 0 (circuit disconnected from GND return). When logic 1 is specified, MOSFET is activated and GND return path is established. The digital output function on Nuvis-3304af series supports sinking current connection. The following diagrams are the suggested wiring for DO:

# DO



Sink

## DIO Channel Definition (Nuvis-3304af)



Pin#	Pin Definition	Pin#	Pin Definition
1	VDD	14	ISO_5V
2	DO_0	15	DO_GND
3	DO_3	16	DO_1
4	DO_GND	17	DO_2
5	DO_5	18	DO_4
6	DO_7	19	DO_6
7	DI_GND	20	DO_GND
8	DI_6	21	DI_GND
9	DI_1	22	DI_5
10	DI_GND	23	DI_3
11	DI_0	24	DI_7
12	DI_2	25	DI_GND
13	DI_4		



# DIO Function Reference (Polling Mode)

## InitDIO

### Syntax

```
BOOL InitDIO(void);
```

### Description

Initialize the DIO function. You should always invoke InitDIO() before write/read any DIO port/channel.

### Parameter

None

### Return Value

Returns TRUE if initialization successes, FALSE if initialization failed.

### Usage

```
BOOL bRet = InitDIO();
```

## DIReadLine

### Syntax

```
BOOL DIReadLine(BYTE ch);
```

### Description

Read a single channel of isolated digital input.

### Parameter

*ch*

BYTE value specifies the DI channel to be read.

For Nuvis-3304af, *ch* should be a value of 0 ~ 7.

### Return Value

The status (TRUE or FALSE) of the specified DI channel.

### Usage

```
BYTE ch=3; //DI channel #3
```

```
BOOL DIChValue = DIReadLine(ch); //read DI channel #3
```

## DIReadPort

### Syntax

```
WORD DIReadPort(void);
```



**Description**

Read the entire isolated digital input port (8 channels).

**Parameter**

None

**Return Value**

A WORD value (0~255) indicates the status of DI port (8 DI channels).

**Usage**

```
WORD  DIPortValue = DIRReadPort ();
```

## DOWriteLine

**Syntax**

```
void DOWriteLine(BYTE ch, BOOL value);
```

**Description**

Write a single channel of isolated digital output.

**Parameter**

*ch*

BYTE value specifies the DO channel to be written.

For Nuvis-3304af, *ch* should be a value of 0 ~ 7.

*value*

BOOL value (TRUE or FALSE) specifies the status of DO channel.

**Return Value**

None

**Usage**

```
BYTE  ch=3; //DI channel #3
```

```
BOOL  DOChValue=TRUE;
```

```
DOWriteLine(ch, DOChValue); //write DO channel #3 as TRUE
```

## DOWritePort

**Syntax**

```
void DOWritePort(WORD value);
```

**Description**

Write the entire isolated digital output port (8 channels).

**Parameter**

*value*

WORD value specifies the status of the DO port. For Nuvis-3304af, *value* should be a value of 0~255.



## Return Value

None

## Usage

```
WORD DOPortValue=0XFF; //11111111b
```

```
DOWritePort(DOPortValue); //write DO port as 11111111b
```



# COS Function Reference

## SetupDICOS

### Syntax

```
BOOL SetupDICOS(COS_INT_SETUP *lpSetup, DWORD cbSetup);
```

### Description

Setup Digital-Input(DI) Change-of-State(COS) interrupt parameters.

### Parameter

*lpSetup* [in]

A pointer to a **COS\_INT\_SETUP** structure that contains the COS configuration information for the DI device.

This data structure contains the following variables:

*portMask*

WORD value specifies the interrupt mask for corresponding channel(s).

*edgeMode*

WORD value specifies that interrupt is generated when level change (set to 0) or on rising/falling edge (set to 1) for the corresponding channel(s).

*edgeType*

WORD value specifies that interrupt is generated on rising edge (set to 0) or falling (set to 1) edge for corresponding channel(s). This value is neglected if *edgeMode* is set to 0 for the corresponding channel(s).

*cbSetup* [in]

The length of the structure, in bytes. The caller must set this member to `sizeof(COS_INT_SETUP)`.

### Return Value

Returns TRUE if setup successes, FALSE if setup failed.

### Usage

```
COS_INT_SETUP  setup;
```

```
memset(&setup, 0, sizeof(setup));
```

```
setup.portMask = 0xff; // enable ch.0~7
```

```
setup.edgeMode = 0; // level
```

```
setup.edgeType = 0x00; // Lo/Hi
```

```
BOOL  bRet = SetupDICOS(&setup, sizeof(setup));
```



## RegisterCallbackDICOS

### Syntax

```
BOOL RegisterCallbackDICOS(COS_INT_CALLBACK callback);
```

### Description

Registers a callback function, which is called when the DICOS interrupt occurred.

### Parameter

*callback* [in]

Specifies the callback function. The prototype for this function is described as follow.

```
void __stdcall callback_func(COS_INT_CALLBACK_ARG* arg);
```

### Return Value

Returns TRUE if setup successes, FALSE if setup failed.

### Usage

```
void __stdcall callback_func(COS_INT_CALLBACK_ARG* arg)
{
    printf("data=0x%02x, flag=0x%02x, seq=%02d\n",
        arg->portData, arg->intrFlag, arg->intrSeq);
}
BOOL bRet = RegisterCallbackDICOS(callback_func);
```

## StartDICOS

### Syntax

```
BOOL StartDICOS(void);
```

### Description

Start DI Change-of-State interrupt.

### Parameter

None

### Return Value

Returns TRUE if start procedure successes, FALSE if start procedure failed.

### Usage

```
BOOL bRet = StartDICOS();
```

## StopDICOS

### Syntax

```
BOOL StopDICOS(void);
```



**Description**

Stop DI Change-of-State interrupt.

**Parameter**

None

**Return Value**

Returns TRUE if stop procedure successes, FALSE if stop procedure failed.

**Usage**

```
BOOL bRet = StopDICOS();
```

**DI-COS Example**

```
#include <stdio.h>
#include <stdlib.h>
#include <windows.h>
#include "WDT_DIO.h"

//Step 0, define a Change-of-State Interrupt callback function
void __stdcall callback_function(COS_INT_CALLBACK_ARG* arg)
{
    printf("data=0x%02x, flag=0x%02x, seq=%02d\n",
        arg->portData, arg->intrFlag, arg->intrSeq);
}

int main(int argc, char* argv[])
{
    //Step 1, initialize DIO library by invoking InitDIO()
    if ( ! InitDIO() )
    {
        printf("InitDIO --> FAILED\n");
        return -1;
    }
    printf("InitDIO --> PASSED\n");

    //Step 2, setup Change-of-State Interrupt mask and level/edge mode
    COS_INT_SETUP setup;

    memset(&setup, 0, sizeof(setup));

    setup.portMask = 0x0f;    // 00001111b, enable ch.0~3
    setup.edgeMode = 0x00;    // generate interrupt on level change
```



```

setup.edgeType = 0x00; // rising/falling edge, only effective when edgeMode = 1

if ( ! SetupDICOS(&setup, sizeof(setup)) )
{
    printf("SetupDICOS --> FAILED\n");
    return -2;
}
printf("SetupDICOS --> PASSED\n");

//Step 3, register the callback function
if ( ! RegisterCallbackDICOS(callback_function) )
{
    printf("RegisterCallbackDICOS --> FAILED\n");
    return -3;
}
printf("RegisterCallbackDICOS --> PASSED\n");

//Step 4, start the DI Change-of-State Interrupt
if ( ! StartDICOS() )
{
    printf("StartDICOS --> FAILED\n");
    return -4;
}
printf("StartDICOS --> PASSED\n");

printf("\npress any key to stop...\n");
system("pause >nul");

//Step 5, stop the DI Change-of-State Interrupt operation
if ( ! StopDICOS() )
{
    printf("StopDICOS --> FAILED\n");
    return -5;
}
printf("StopDICOS --> PASSED\n");

printf("\npress any key to exit...\n");
system("pause >nul");
return 0;
}

```



# DTIO Function Reference

## SetupDTIO

### Syntax

```
BOOL SetupDTIO(DTIO_SETUP *lpSetup, DWORD cbSetup);
```

### Description

Setup timing correlation of trigger source input (DI) and pulse target output (DO) for Deterministic Trigger I/O function.

### Parameter

*lpSetup* [in]

A pointer to a **DTIO\_SETUP** structure that contains the DTIO configuration. You can invoke this function multiple times for specifying the timing correlation between single DI channel and multiple DO channels.

This data structure contains the following variables:

*trigMode*

BYTE value specifies which condition of DI input shall be recognized as a trigger event.

*trigSrcDI*

BYTE value specifies a DI channel used as trigger source input. For Nuvis-3304af, *trigSrcDI* should be a value of 0 ~ 7.

*pulseTgtDO*

BYTE value specifies a DO channel used as pulse target output. For Nuvis-3304af, *pulseTgtDO* should be a value of 0 ~ 7.

*pulseExtra*

BYTE value specifies a combination of flags for additional DTIO configuration. Current available flag is DTIO\_INIT\_HIGN only (output a active-low pulse).

*pulseDelay*

DWORD value specifies the tick count of output pulse delay. *pulseDelay* should be a value of 2 ~ 2147483647. The absolute delay time is *pulseDelay* value multiplied by timing unit (25us by default).

If *pulseDelay* is 0, Deterministic Trigger Fan-out (DTFO) mode is configured. Output pulse is generated immediately when trigger event occurred.

*pulseWidth*

DWORD value specifies the tick count of output pulse width. *pulseWidth* should be a value of 1 ~ 2147483647. The absolute pulse width is *pulseWidth* value multiplied by timing unit (25us by default).





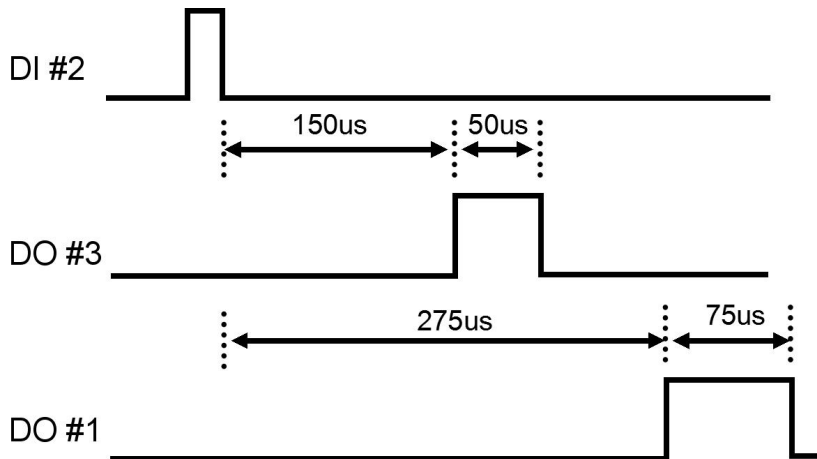
*cbSetup* [in]

The length of the structure, in bytes. The caller must set this member to `sizeof(DTIO_SETUP)`.

### Return Value

Returns TRUE if setup successes, FALSE if setup failed.

### Usage



For specifying the above timing correlation between DI#2, DO#3 and DO#1, please refer the reference code below.

```
DTIO_SETUP    DO3_setup;
DTIO_SETUP    DO1_setup;
BOOL          bRet;
```

```
memset(&DO3_setup, 0, sizeof(DTIO_SETUP));
memset(&DO1_setup, 0, sizeof(DTIO_SETUP));
```

```
//Setup DO#3 pulse output
```

```
DO3_setup.trigMode = 3;    //Specify the falling edge on DI as trigger event.
DO3_setup.trigSrcDI = 2;  //Specify DI#2 as the trigger source DI channel.
DO3_setup.pulseTgtDO = 3; //Specify DO#3 as the pulse output channel.
DO3_setup.pulseExtra = 0; //Specify a active-high pulse output (Default).
DO3_setup.pulseDelay = 6; //Specify a delay of 6 x 25 = 150us after a trigger event
                          //occurred.
DO3_setup.pulseWidth = 2; //Specify a pulse width of 2 x 25 = 50us.
```

```
//Setup DO#1 pulse output
```

```
DO1_setup.trigMode = 3;    //Specify the falling edge on DI as trigger event.
DO1_setup.trigSrcDI = 2;  //Specify DI#2 as the trigger source DI channel.
DO1_setup.pulseTgtDO = 1; //Specify DO#1 as the pulse output channel.
```



```

DO1_setup.pulseExtra = 0;    //Specify a active-high pulse output (Default).
DO1_setup.pulseDelay = 11;  //Specify a delay of 11 x 25 = 275 after a trigger event
                              //occurred.
DO1_setup.pulseWidth = 3;   //Specify a pulse width of 3 x 25 = 75us.

//Invoke SetupDTIO twice to setup timing correlation for DO#3 and DO#1
bRet = SetupDTIO (&DO3_setup, sizeof(DO3_setup));
bRet = SetupDTIO (&DO1_setup, sizeof(DO1_setup));

```

## StartDTIO

### Syntax

```
BOOL StartDTIO(void);
```

### Description

Start Deterministic Trigger I/O operation.

### Parameter

None

### Return Value

Returns TRUE if start procedure successes, FALSE if start procedure failed.

### Usage

```
BOOL bRet = StartDTIO();
```

## StopDTIO

### Syntax

```
BOOL StopDTIO(void);
```

### Description

Stop Deterministic Trigger I/O operation.

### Parameter

None

### Return Value

Returns TRUE if stop procedure successes, FALSE if stop procedure failed.

### Usage

```
BOOL bRet = StopDTIO();
```

## SetUnitDTIO

### Syntax

```
BOOL SetUnitDTIO(WORD unit, int delta);
```

### Description



Specify the unit of timing for DTIO. This function can be skipped if you want to use the default setting (25us).

### Parameter

*unit*

WORD value specifies the timing unit in the unit of micro-second. *unit* should be a value of 25 ~ 2500.

*delta*

Integer value specifies the fine-tuning factor for timing unit. The value of +/- 1 can increase/decrease 0.04us for the timing unit. Default value of *delta* is 0.

### Return Value

Returns TRUE if start procedure successes, FALSE if start procedure failed.

### Usage

```
BOOL bRet = SetUnitDTIO(25, 0); //Specify a 25us timing unit.
```

## GetUnitDTIO

### Syntax

```
WORD GetUnitDTIO(void);
```

### Description

Get the current setting for timing unit in micro-second.

### Parameter

None

### Return Value

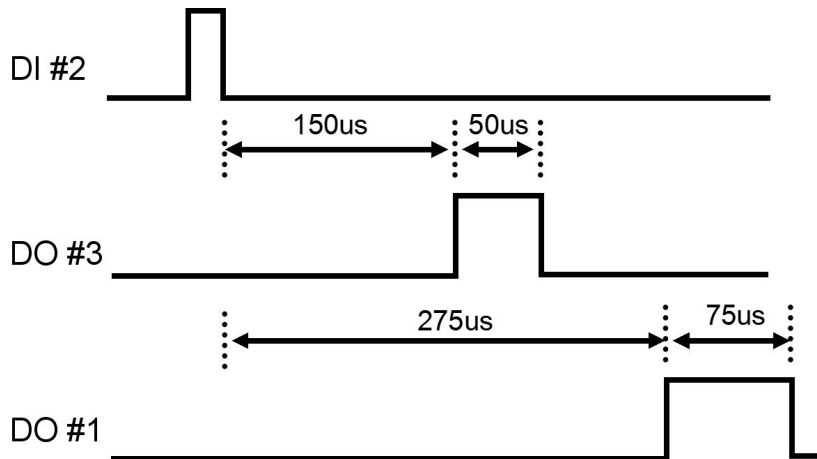
WORD value for the current timing unit.

### Usage

```
WORD wUnit = GetUnitDTIO();
```

## DTIO Example 1 -- Deterministic Timing between DI & DO

This example demonstrates the usage of using DI channel#2 to accept an trigger signal (at falling edge) and then generating two pulse outputs on DO channel#3 and DO channel#1 according to different delay and pulse width settings.



```
#include <stdio.h>
#include <stdlib.h>
#include <windows.h>
#include "WDT_DIO.h"
```

```
int main(int argc, char* argv[])
{
    //Step 1, initialize DIO library
    if (! InitDIO() )
    {
        printf("InitDIO --> FAILED\n");
        return -1;
    }
    printf("InitDIO --> PASSED\n");

    //Step 2, setup a timing unit of 25us.
    //This step can be skipped for default setting (25us) is applied.
    if (! SetUnitDTIO(25, 0) )
    {
        printf("SetUnitDTIO--> FAILED\n");
        return -2;
    }
    printf("SetUnitDTIO--> PASSED\n");
```



```

//Step 3, setup Deterministic Trigger I/O for DO#3 and DO#1
DTIO_SETUP   DO3_setup;
DTIO_SETUP   DO1_setup;

memset(&DO3_setup, 0, sizeof(DTIO_SETUP ));
memset(&DO1_setup, 0, sizeof(DTIO_SETUP ));

//Setup DO#3 pulse output
DO3_setup.trigMode = 3;      //Specify the falling edge on DI as trigger event.
DO3_setup.trigSrcDI = 2;    //Specify DI#2 as the trigger source DI channel.
DO3_setup.pulseTgtDO = 3;   //Specify DO#3 as the pulse output channel.
DO3_setup.pulseExtra = 0;   //Specify a active-high pulse output (Default).
DO3_setup.pulseDelay = 6;   //Specify a delay of 6 x 25 = 150us after a trigger event
                             //occurred.
DO3_setup.pulseWidth = 2;   //Specify a pulse width of 2 x 25 = 50us.

//Setup DO#1 pulse output
DO1_setup.trigMode = 3;      //Specify the falling edge on DI as trigger event.
DO1_setup.trigSrcDI = 2;    //Specify DI#2 as the trigger source DI channel.
DO1_setup.pulseTgtDO = 1;   //Specify DO#1 as the pulse output channel.
DO1_setup.pulseExtra = 0;   //Specify a active-high pulse output (Default).
DO1_setup.pulseDelay = 11;  //Specify a delay of 11 x 25 = 275 after a trigger event
                             //occurred.
DO1_setup.pulseWidth = 3;   //Specify a pulse width of 3 x 25 = 75us.

if ( ! SetupDTIO(&DO3_setup, sizeof(DO3_setup)) )
{
    printf("SetupDTIO for DO#3 --> FAILED\n");
    return -3;
}
printf("SetupDTIO for DO3 --> PASSED\n");

if ( ! SetupDTIO(&DO1_setup, sizeof(DO1_setup)) )
{
    printf("SetupDTIO for DO#1 --> FAILED\n");
    return -3;
}
printf("SetupDTIO for DO#1 --> PASSED\n");

```



```

//Step 4, start Deterministic Trigger I/O
if ( ! StartDTIO() )
{
    printf("StartDTIO --> FAILED\n");
    return -4;
}
printf("StartDTIO --> PASSED\n");

printf("\npress any key to stop...\n");
system("pause >nul");

//Step 5, stop Deterministic Trigger I/O
if ( ! StopDTIO() )
{
    printf("StopDTIO --> FAILED\n");
    return -5;
}
printf("StopDTIO --> PASSED\n");

printf("\npress any key to exit...\n");
system("pause >nul");
return 0;
}

```

## DTIO Example 2 -- Infinite Pulse Train

This example demonstrates the usage of generating a infinite pulse train on DO channel#0 with interval of 25us and pulse width of 25us.

```

#include <stdio.h>
#include <stdlib.h>
#include <windows.h>
#include "WDT_DIO.h"

int main(int argc, char* argv[])
{
    //Step 1, initialize DIO library
    if ( ! InitDIO() )
    {

```



```

    printf("InitDIO --> FAILED\n");
    return -1;
}
printf("InitDIO --> PASSED\n");

//Step 2, setup Deterministic Trigger I/O
DTIO_SETUP  setup;

memset(&setup, 0, sizeof(setup));

setup.trigMode = 1;      // 1: always triggered mode (infinite pulse)
setup.pulseTgtDO = 0;   // DO channel 0
setup.pulseDelay = 1;   // tick count, 25us delay
setup.pulseWidth = 1;  // tick count, 25us width

if ( ! SetupDTIO(&setup, sizeof(setup)) )
{
    printf("SetupDTIO --> FAILED\n");
    return -2;
}
printf("SetupDTIO --> PASSED\n");

//Step 3, start Deterministic Trigger I/O
if ( ! StartDTIO() )
{
    printf("StartDTIO --> FAILED\n");
    return -3;
}
printf("StartDTIO --> PASSED\n");

printf("\npress any key to stop...\n");
system("pause >nul");

//Step 4, stop Deterministic Trigger I/O
if ( ! StopDTIO() )
{
    printf("StopDTIO --> FAILED\n");
    return -4;
}
printf("StopDTIO --> PASSED\n");

```



```
printf("\npress any key to exit...\n");  
system("pause >nul");  
return 0;  
}
```





# Deterministic Trigger Fan-Out Function Reference

## SetupDTFO

### Syntax

```
BOOL SetupDTFO(DTFO_SETUP *lpSetup, DWORD cbSetup);
```

### Description

Setup timing correlation of trigger source input (DI) and pulse target output (DO) for Deterministic Trigger Fan-Out function.

### Parameter

*lpSetup* [in]

A pointer to a **DTFO\_SETUP** structure that contains the DTFO configuration. You can invoke this function multiple times for specifying the timing correlation between single DI channel and multiple DO channels.

This data structure contains the following variables:

*trigMode*

BYTE value specifies which condition of DI input shall be recognized as a trigger event.

- 1: reserved
- 2: triggered on rising edge
- 3: triggered on falling edge

*trigSrcDI*

BYTE value specifies a DI channel used as trigger source input. For Nuvis-3304af, *trigSrcDI* should be a value of 0 ~ 7.

*pulseTgtDO*

BYTE value specifies a DO channel used as pulse target output. For Nuvis-3304af, *pulseTgtDO* should be a value of 0 ~ 7.

*pulseExtra*

BYTE value specifies a combination of flags for additional DTFO configuration. Current available flag is DTFO\_INIT\_HIGN only (output a active-low pulse).

*pulseTag*

DWORD value. Not used for DTFO function.

*pulseWidth*

DWORD value specifies the tick count of output pulse width. *pulseWidth* should be a value of 1 ~ 2147483647. The absolute pulse width is *pulseWidth* value multiplied by timing unit (25us by default).

*cbSetup* [in]

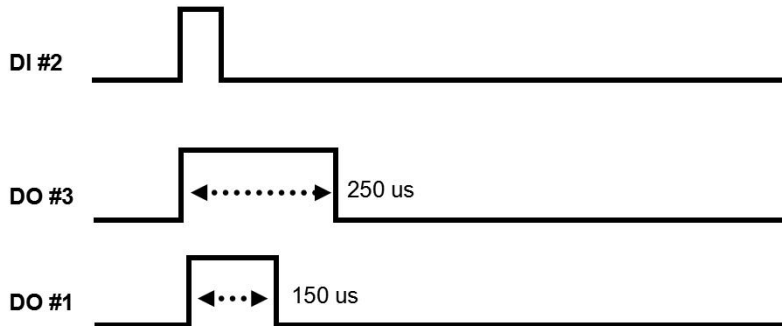


The length of the structure, in bytes. The caller must set this member to `sizeof(DTFO_SETUP)`.

### Return Value

Returns TRUE if setup successes, FALSE if setup failed.

### Usage



For specifying the above timing correlation between DI#2, DO#3 and DO#1, please refer the reference code below.

```
DTFO_SETUP  DO3_setup;
DTFO_SETUP  DO1_setup;
BOOL        bRet;
```

```
memset(&DO3_setup, 0, sizeof(DTFO_SETUP ));
memset(&DO1_setup, 0, sizeof(DTFO_SETUP ));
```

```
//Setup DO#3 pulse output
```

```
DO3_setup.trigMode = 3;    //Specify the falling edge on DI as trigger event.
DO3_setup.trigSrcDI = 2;  //Specify DI#2 as the trigger source DI channel.
DO3_setup.pulseTgtDO = 3; //Specify DO#3 as the pulse output channel.
DO3_setup.pulseExtra = 0; //Specify a active-high pulse output (Default).
DO3_setup.pulseTag= 0;    //pulseTag is not used for DTFO.
DO3_setup.pulseWidth = 10; //Specify a pulse width of 10 x 25 = 250us.
```

```
//Setup DO#1 pulse output
```

```
DO1_setup.trigMode = 3;    //Specify the falling edge on DI as trigger event.
DO1_setup.trigSrcDI = 2;  //Specify DI#2 as the trigger source DI channel.
DO1_setup.pulseTgtDO = 1; //Specify DO#1 as the pulse output channel.
DO1_setup.pulseExtra = 0; //Specify a active-high pulse output (Default).
DO1_setup.pulseTag = 0;    //pulseTag is not used for DTFO.
DO1_setup.pulseWidth = 6; //Specify a pulse width of 6 x 25 = 150us.
```

```
//Invoke SetupDTFO twice to setup timing correlation for DO#3 and DO#1
bRet = SetupDTFO (&DO3_setup, sizeof(DO3_setup));
bRet = SetupDTFO (&DO1_setup, sizeof(DO1_setup));
```

## StartDTFO

### Syntax

```
BOOL StartDTFO(void);
```

### Description

Start Deterministic Trigger Fan-Out operation.

### Parameter

None

### Return Value

Returns TRUE if start procedure successes, FALSE if start procedure failed.

### Usage

```
BOOL bRet = StartDTFO();
```

## StopDTFO

### Syntax

```
BOOL StopDTFO(void);
```

### Description

Stop Deterministic Trigger Fan-Out operation.

### Parameter

None

### Return Value

Returns TRUE if stop procedure successes, FALSE if stop procedure failed.

### Usage

```
BOOL bRet = StopDTFO();
```

## SetUnitDTFO

### Syntax

```
BOOL SetUnitDTFO(WORD unit, int delta);
```

### Description

Specify the unit of timing for DTFO. This function can be skipped if you want to use the default setting (25us).

### Parameter

*unit*



WORD value specifies the timing unit in the unit of micro-second. *unit* should be a value of 25 ~ 2500.

*delta*

Integer value specifies the fine-tuning factor for timing unit. The value of +/- 1 can increase/decrease 0.04us for the timing unit. Default value of *delta* is 0.

### Return Value

Returns TRUE if start procedure successes, FALSE if start procedure failed.

### Usage

```
BOOL bRet = SetUnitDTFO(25, 0); //Specify a 25us timing unit.
```

## GetUnitDTFO

### Syntax

```
WORD GetUnitDTFO(void);
```

### Description

Get the current setting for timing unit in micro-second.

### Parameter

None

### Return Value

WORD value for the current timing unit.

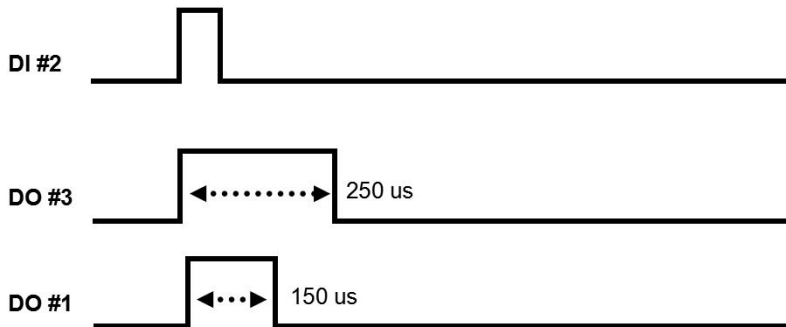
### Usage

```
WORD wUnit = GetUnitDTFO();
```



## DTFO Example 1 -- Deterministic Trigger Fan-Out

This example demonstrates the usage of using DI channel#2 to accept an trigger signal (at falling edge) and then generating two trigger fan-out signals on DO channel#3 and DO channel#1 according to different delay and pulse width settings.



```
#include <stdio.h>
#include <stdlib.h>
#include <windows.h>
#include "WDT_DIO.h"

int main(int argc, char* argv[])
{
    //Step 1, initialize DIO library
    if (! InitDIO() )
    {
        printf("InitDIO --> FAILED\n");
        return -1;
    }
    printf("InitDIO --> PASSED\n");

    //Step 2, setup a timing unit of 25us.
    //This step can be skipped for default setting (25us) is applied.
    if (! SetUnitDTFO(25, 0) )
    {
        printf("SetUnitDTFO--> FAILED\n");
        return -2;
    }
    printf("SetUnitDTFO--> PASSED\n");

    //Step 3, setup Deterministic Trigger Fan-Out for DO#3 and DO#1
```



```

DTFO_SETUP DO3_setup;
DTFO_SETUP DO1_setup;

memset(&DO3_setup, 0, sizeof(DTFO_SETUP ));
memset(&DO1_setup, 0, sizeof(DTFO_SETUP ));

//Setup DO#3 pulse output
DO3_setup.trigMode = 3;      //Specify the falling edge on DI as trigger event.
DO3_setup.trigSrcDI = 2;    //Specify DI#2 as the trigger source DI channel.
DO3_setup.pulseTgtDO = 3;  //Specify DO#3 as the pulse output channel.
DO3_setup.pulseExtra = 0;  //Specify a active-high pulse output (Default).
DO3_setup.pulseTag = 0;    //pulseTag is not used for DTFO.
DO3_setup.pulseWidth = 10; //Specify a pulse width of 10 x 25 = 250us.

//Setup DO#1 pulse output
DO1_setup.trigMode = 3;      //Specify the falling edge on DI as trigger event.
DO1_setup.trigSrcDI = 2;    //Specify DI#2 as the trigger source DI channel.
DO1_setup.pulseTgtDO = 1;  //Specify DO#1 as the pulse output channel.
DO1_setup.pulseExtra = 0;  //Specify a active-high pulse output (Default).
DO1_setup.pulseTag = 0;    //pulseTag is not used for DTFO.
DO1_setup.pulseWidth = 6;  //Specify a pulse width of 6 x 25 = 150us.

if ( ! SetupDTFO(&DO3_setup, sizeof(DO3_setup)) )
{
    printf("SetupDTFO for DO#3 --> FAILED\n");
    return -3;
}
printf("SetupDTFO for DO3 --> PASSED\n");

if ( ! SetupDTFO(&DO1_setup, sizeof(DO1_setup)) )
{
    printf("SetupDTFO for DO#1 --> FAILED\n");
    return -3;
}
printf("SetupDTFO for DO#1 --> PASSED\n");

//Step 4, start Deterministic Trigger Fan-Out
if ( ! StartDTFO() )
{
    printf("StartDTFO --> FAILED\n");
}

```



```
        return -4;
    }
    printf("StartDTFO --> PASSED\n");

    printf("\npress any key to stop...\n");
    system("pause >nul");

    //Step 5, stop Deterministic Trigger Fan-Out
    if ( ! StopDTFO() )
    {
        printf("StopDTFO --> FAILED\n");
        return -5;
    }
    printf("StopDTFO --> PASSED\n");

    printf("\npress any key to exit...\n");
    system("pause >nul");
    return 0;
}
```



# Appendix B Using Per-Port PoE On/Off Control

Nuvis-3304af series supports a unique feature of per-port power on/off control for each of its PoE ports. With provided function APIs, users can turn on or turn off the power of each PoE port manually for fault-recovery or device power reset purpose.

The function APIs are encapsulated in Neosys WDT\_DIO driver package. Please following the instructions in **Appendix A - Install WDT and DIO Library** to install the driver package in prior to program per-port PoE on/off control function.

## Per-Port PoE On/Off Control Function Reference

### GetStatusPoEPort

#### Syntax

```
BYTE GetStatusPoEPort (Byte port);
```

#### Description

Get the current on/off status of designated PoE port.

#### Parameter

*port*

BYTE value specifies the PoE port. For Nuvis-3304af, *port* should be a value of 1 ~ 4. You can see the port# on the rear panel of Nuvis-3304af.

#### Return Value

BYTE value for on/off status. 0: Disabled (off), 1: Enabled (on)

#### Usage

```
BYTE bEnabled = GetStatusPoEPort (1); //Get on/off status of PoE Port#1
```

### EnablePoEPort

#### Syntax

```
BOOL EnablePoEPort (BYTE port);
```

#### Description

Turn on PoE power of designated PoE port.

#### Parameter

*port*

BYTE value specifies the PoE port. For Nuvis-3304af, *port* should be a value





of 1 ~ 4. You can see the port# on the rear panel of Nuvis-3304af.

**Return Value**

Returns TRUE if start procedure successes, FALSE if start procedure failed.

**Usage**

```
BOOL bRet = EnablePoEPort (1); //Turn on PoE Port#1
```

## DisablePoEPort

**Syntax**

```
BOOL DisablePoEPort (BYTE port);
```

**Description**

Turn off PoE power of designated PoE port.

**Parameter**

*port*

BYTE value specifies the PoE port. For Nuvis-3304af, *port* should be a value of 1 ~ 4. You can see the port# on the rear panel of Nuvis-3304af.

**Return Value**

Returns TRUE if start procedure successes, FALSE if start procedure failed.

**Usage**

```
BOOL bRet = DisablePoEPort (1); //Turn off PoE Port#1
```