ET970 Series

Intel® 7th Gen. Core™ / Xeon® E3
COM Express Type 6Module

User's Manual

Version 1.1
(Oct 2019)

Compliance

CE

This product has passed CE Class B tests for environmental specifications and limits. This product is in accordance with the directives of the European Union (EU). In a domestic environment, this product may cause radio interference in which case users may be required to take adequate measures.

FC

This product has been tested and found to comply with the limits for a Class B device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with manufacturer's instructions, may cause harmful interference to radio communications.

WEEE

This product must not be disposed of as normal household waste, in accordance with the EU directive of for waste electrical and electronic equipment (WEEE - 2012/19/EU). Instead, it should be disposed of by returning it to a municipal recycling collection point. Check local regulations for disposal of electronic products.

Green IBASE

This product is compliant with the current RoHS restrictions and prohibits use of the following substances in concentrations exceeding 0.1% by weight (1000 ppm) except for cadmium, limited to 0.01% by weight (100 ppm).

- Lead (Pb)
- Mercury (Hg)
- Cadmium (Cd)
- Hexavalent chromium (Cr6+)
- Polybrominated biphenyls (PBB)
- Polybrominated diphenyl ether (PBDE)

Important Safety Information

Carefully read the precautions before using the board.

Environmental conditions:

- Use this product in environments with ambient temperatures between 0˚C and 60˚C.
- Do not leave this product in an environment where the storage temperature may be below -20° C or above 80° C. To prevent from damages, the product must be used in a controlled environment.

Care for your IBASE products:

- Before cleaning the PCB, unplug all cables and remove the battery.
- Clean the PCB with a circuit board cleaner or degreaser, or use cotton swabs and alcohol.
- Vacuum the dust with a computer vacuum cleaner to prevent the fan from being clogged.

WARNING

Attention during use:

- Do not use this product near water.
- Do not spill water or any other liquids on this product.
- Do not place heavy objects on the top of this product.

Anti-static precautions

- Wear an anti-static wrist strap to avoid electrostatic discharge.
- Place the PCB on an anti-static kit or mat.
- Hold the edges of PCB when handling.
- Touch the edges of non-metallic components of the product instead of the surface of the PCB.
- Ground yourself by touching a grounded conductor or a grounded bit of metal frequently to discharge any static.

CAUTION

Danger of explosion if the internal lithium-ion battery is replaced by an incorrect type. Replace only with the same or equivalent type recommended by the manufacturer. Dispose of used batteries according to the manufacturer's instructions or recycle them at a local recycling facility or battery collection point.

Warranty Policy

- **IBASE standard products:**

  24-month (2-year) warranty from the date of shipment. If the date of shipment cannot be ascertained, the product serial numbers can be used to determine the approximate shipping date.

- **3rd-party parts:**

  12-month (1-year) warranty from delivery for the 3rd-party parts that are not manufactured by IBASE, such as CPU, CPU cooler, memory, storage devices, power adapter, panel and touchscreen.

* PRODUCTS, HOWEVER, THAT FAIL DUE TO MISUSE, ACCIDENT, IMPROPER INSTALLATION OR UNAUTHORIZED REPAIR SHALL BE TREATED AS OUT OF WARRANTY AND CUSTOMERS SHALL BE BILLED FOR REPAIR AND SHIPPING CHARGES.

Technical Support & Services

1. Visit the IBASE website at www.ibase.com.tw to find the latest information about the product.
2. If you need any further assistance from your distributor or sales representative, prepare the following information of your product and elaborate upon the problem.
   - Product model name
   - Product serial number
   - Detailed description of the problem
   - The error messages in text or in screenshots if there is any
   - The arrangement of the peripherals
   - Software in use (such as OS and application software, including the version numbers)
3. If repair service is required, you can download the RMA form at http://www.ibase.com.tw/english/Supports/RMAService/. Fill out the form and contact your distributor or sales representative.

Table of Contents

This page is intentionally left blank.

Chapter 1
General Information

The information provided in this chapter includes:

- Features
- Packing List
- Optional Accessories
- Specifications
- Block Diagram
- Board Overview
- Board Dimensions

1.1 Introduction

ET970 is a COM Express module based on the platform of Intel® 7th Gen. Core™ / Xeon® E3 with Intel® QM175 / CM238 chipset integrated. It comes with type 6 pinouts, and complies with the PICMG COM.0 specifications. ET970 features two DDR4 dual channel memory slots and outputs signals for DVI, HDMI, DisplayPort, and LVDS interface connection. It is able to be operated at the ambient operating temperature ranging from 0 ~ 60 °C, and even from -20 ~ 80 °C for storage.

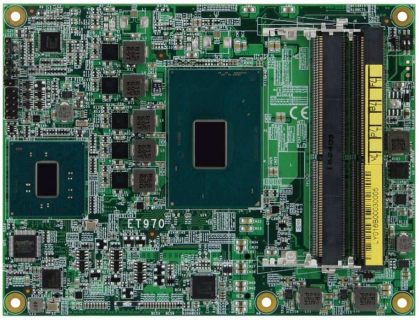


Photo of ET970

1.2 Features

- COM Express module type 6 with Intel® 7th Gen. Core™ / Xeon® E3 processor
- 2 x DDR4-2400 SO-DIMM, Max. 32 GB
- 1 x Intel® PCIe GbE LAN
- 8 x USB 2.0, 4 x USB 3.0, 2 x COM, 4 x SATA III
- 3 independent displays and display output signals for interfaces of DVI-D, HDMI, DisplayPort and LVDS
- 8 x PCIe (x1), 1 x PCIe (x16)
- Configurable watchdog timer, digital I/O, mSATA, TPM 2.0

1.3 Packing List

Your ET970 package should include the items listed below. If any of the items below is missing, contact the distributor or dealer from whom you purchased the product.

- ET970 COM Express Module x 1
- Disk x 1
  (including chipset drivers and flash memory utility)
- This User's Manual x 1

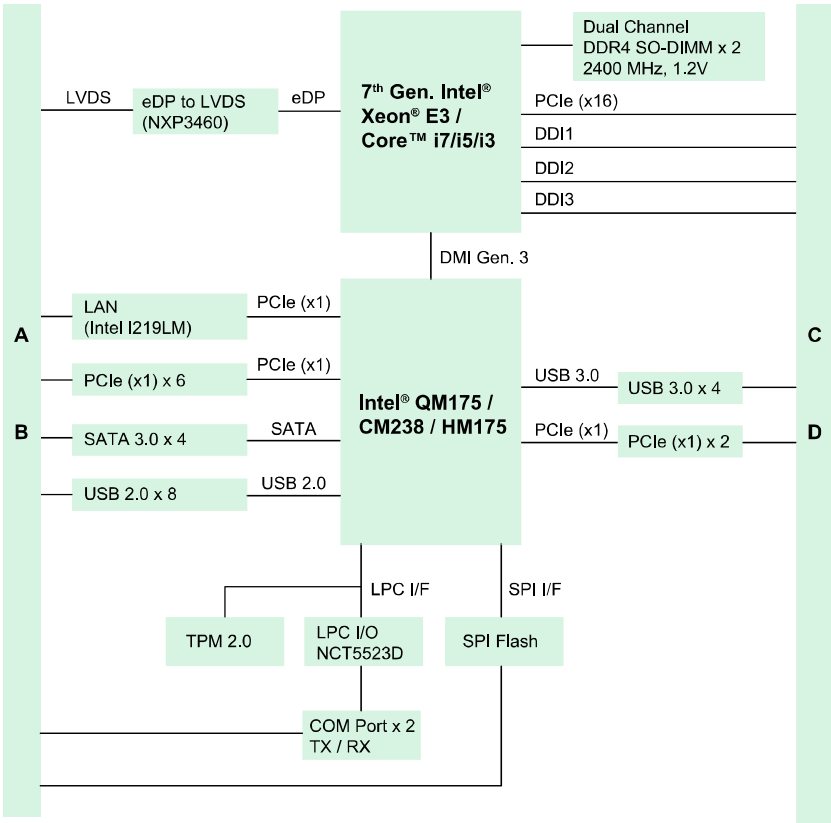
1.4 Specifications

Product Name	ET970 Series
System	
Operating System	Windows 10 (64-bit)
CPU & Chipset	• **ET970K-i7:** Intel® 7th Gen. Core™ i7-7820EQ (3~3.7GHz) with QM175 PCH • **ET970K-i5:** Intel® 7th Gen. Core™ i5-7440EQ (2.9 ~ 3.6 GHz) with QM175 PCH • **ET970K-i3:** Intel® 7th Gen. Core™ i3-7100E (2.9 GHz) with HM175 PCH • **ET970K-X3G:** Intel® Xeon® E3-1505M V6 (3 ~ 4 GHz) with CM238 PCH
Memory	2 x DDR4-2400 SO-DIMM, dual channel, expandable up to 32 GB
Graphics	Intel® 7th Gen. Core™ H-series integrated graphics
Network	Intel® I219LM PHY
Super I/O	Nuvoton NCT5523D
Power Requirement	5V / 12V DC-In
TPM	2.0

Watchdog Timer	Yes (256 segments, 0, 1, 2…255 sec / min)
BIOS	AMI BIOS
H/W Monitor	Yes
Dimensions	95 x 125 mm (3.74” x 4.92”)
RoHS	Yes
Certification	CE, FCC Class B, LVDS
I/O Ports	
Display	• 1 x HDMI • 1 x Display Port • 1 x DVI-D • 1 x 24-bit dual channel LVDS
LAN	1 x RJ45 GbE LAN
USB	• 4 x USB 3.0 • 8 x USB 2.0
Serial	• COM1: RS-232/422/485 • COM2: RS-232 (full functions) • COM3 & COM4: RS-232 (TX and RX)
SATA	4 x SATA 3.0
Audio	Line-In, Line-Out, and Mic-In
Expansion Slots	• 8 x PCIe slot (x1) • 1 x PCIe slot (x16)
Environment	
Temperature	• Operation: 0 ~ 60 °C (32 ~ 140 °F) • Storage: -20 ~ 80 °C (-4 ~ 176 °F)
Relative Humidity	10 ~ 90 %, non-condensing

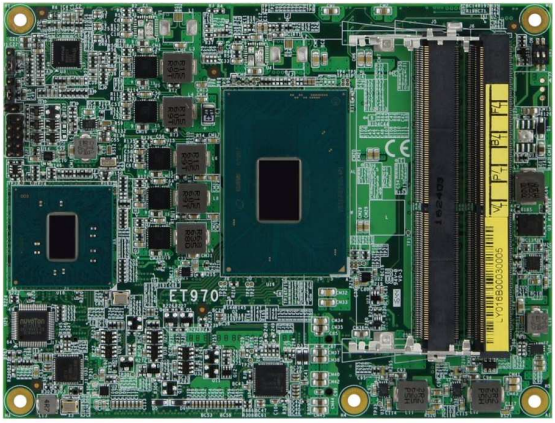
All specifications are subject to change without prior notice.

1.5 Block Diagram

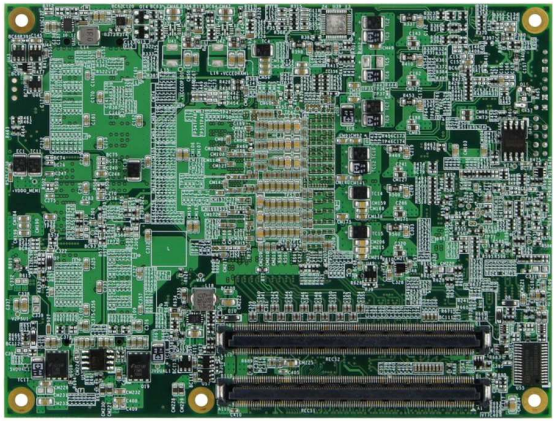


1.6 Overview

Top View



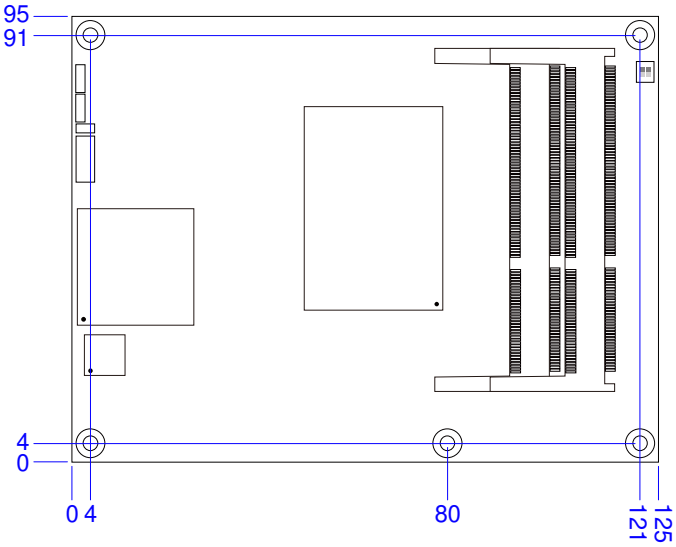
Bottom View



Photos of ET970

* The photos above are for reference only. Some minor components may differ.

1.7 Dimensions



This page is intentionally left blank.

Chapter 2
Hardware Configuration

This section provides information on jumper settings and connectors on the ET970 in order to set up a workable system. On top of that, you will also need to install crucial pieces such as the CPU and the memory before using the product. The topics covered are:

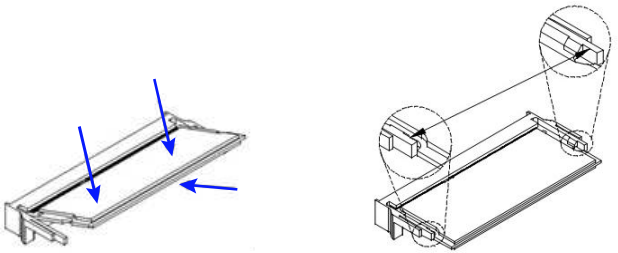
- Essential installations before you begin
- Jumper and connector locations
- Jumper settings and information of connectors

2.1 Essential Installations Before You Begin

Follow the instructions below to install the memory.

2.1.1 Installing the Memory

If you need to replace or install a memory module, locate the memory slot on the board and perform the following steps:



1. Align the key of the memory module with that on the memory slot and insert the module slantwise.
2. Gently push the module in an upright position until the clips of the slot close to hold the module in place when the module touches the bottom of the slot.

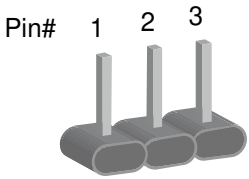
To remove the module, press the clips outwards with both hands, and the module will pop-up.

2.2 Setting the Jumpers

Set up and configure your ET970 by using jumpers for various settings and features according to your needs and applications. Contact your supplier if you have doubts about the best configuration for your use.

2.2.1 How to Set Jumpers

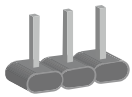
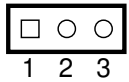
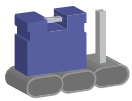
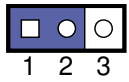
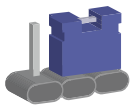
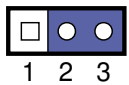
Jumpers are short-length conductors consisting of several metal pins with a non-conductive base mounted on the circuit board. Jumper caps are used to have the functions and features enabled or disabled. If a jumper has 3 pins, you can connect either PIN1 to PIN2 or PIN2 to PIN3 by shorting.



A 3-pin jumper

A jumper cap

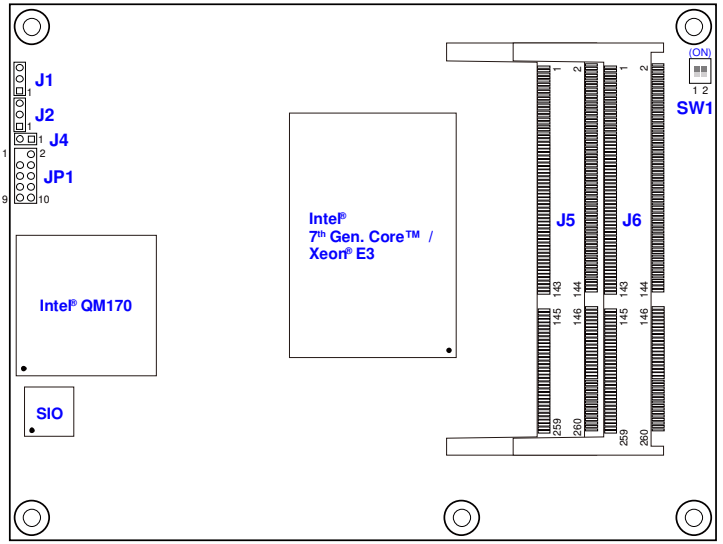
Refer to the illustration below to set jumpers.

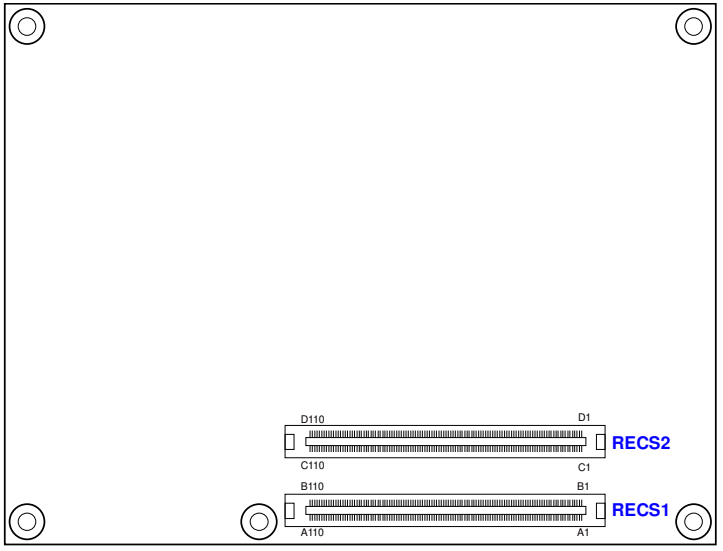
Pin closed	Oblique view	Schematic illustration in the manual
Open		
1-2		
2-3		

When two pins of a jumper are encased in a jumper cap, this jumper is **closed**, i.e. turned **On**.

When a jumper cap is removed from two jumper pins, this jumper is **open**, i.e. turned **Off**.

2.3 Jumper & Connector Locations





Board diagram of ET970

2.4 Jumpers & Connector Quick Reference

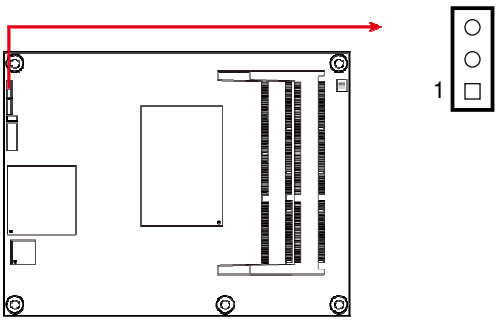
Jumpers:

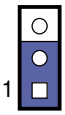
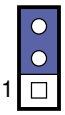
Function	Jumper Name	Page
ME Register Clearance	J1	13
CMOS Data Clearance	J2	14
Factory Use Only	J4, JP1	- -

Connectors:

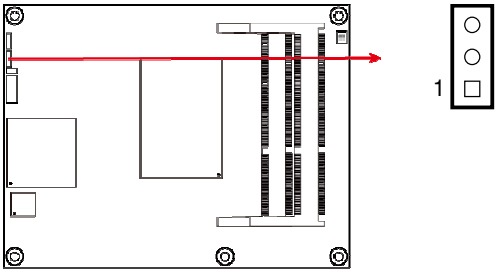
Function	Jumper Name	Page
DDR4 SO-DIMM Slots	J5, J6	14
PCIe (x16) Bifurcation Selection	SW1	15
COM Express Module Type 6 Connector	RECS1, RECS2	16

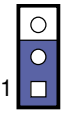
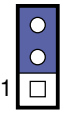
2.4.1 ME Register Clearance (J1)



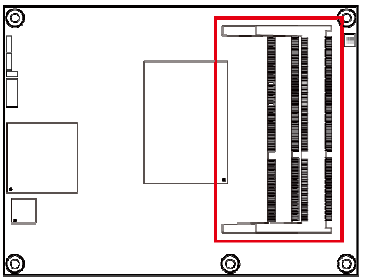
Function	Pin closed	Illustration
Normal (default)	1-2	
Clear ME	2-3	

2.4.2 CMOS Data Clearance (J2)

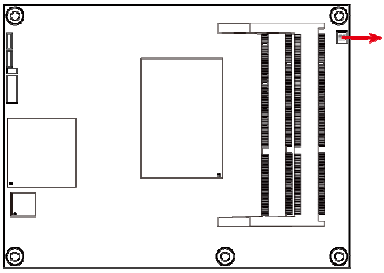


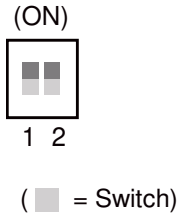
Function	Pin closed	Illustration
Normal (default)	1-2	
Clear CMOS	2-3	

2.4.3 DDR4 SO-DIMM Slots (J5, J6)



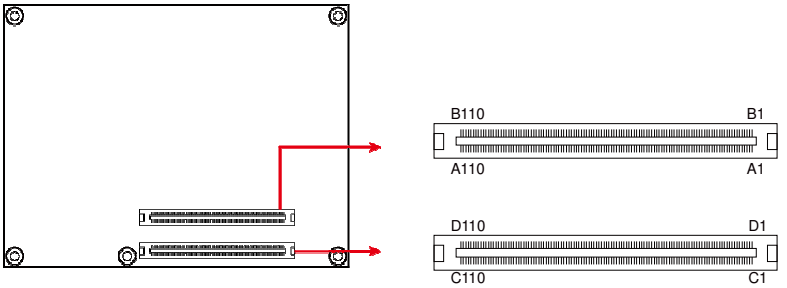
2.4.4 PCIe (x16) Bifurcation Selection (SW1)





Function	Pin closed	Illustration
1 x PCIe (x16) (default)	Pin 1: **Off** Pin2: **Off**	(ON) 1 2
2 x PCIe (x8)	Pin 1: **On** Pin2: **Off**	(ON) 1 2
1 x PCIe (x8) 2 x PCIe (x4)	Pin 1: **On** Pin2: **On**	(ON) 1 2

2.4.5 COM Express Module Type 6 Connector (RECS1, RECS2)



Row A		Row B		Row C		Row D	
Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
A1	GND (FIXED)	B1	GND (FIXED)	C1	GND (FIXED)	D1	GND (FIXED)
A2	GBE0_MDI3-	B2	GBE0_ACT#	C2	GND	D2	GND
A3	GBE0_MDI3+	B3	LPC_FRAME#	C3	USB_SSRX0-	D3	USB_SSTX0-
A4	GBE0_LINK100#	B4	LPC_AD0	C4	USB_SSRX0+	D4	USB_SSTX0+
A5	GBE0_LINK1000#	B5	LPC_AD1	C5	GND	D5	GND
A6	GBE0_MDI2-	B6	LPC_AD2	C6	USB_SSRX1-	D6	USB_SSTX1-
A7	GBE0_MDI2+	B7	LPC_AD3	C7	USB_SSRX1+	D7	USB_SSTX1+
A8	GBE0_LINK#	B8	LPC_DRQ0#	C8	GND	D8	GND
A9	GBE0_MDI1-	B9	LPC_DRQ1#	C9	USB_SSRX2-	D9	USB_SSTX2-
A10	GBE0_MDI1+	B10	LPC_CLK	C10	USB_SSRX2+	D10	USB_SSTX2+
A11	GND (FIXED)	B11	GND (FIXED)	C11	GND (FIXED)	D11	GND (FIXED)
A12	GBE0_MDI0-	B12	PWRBTN#	C12	USB_SSRX3-	D12	USB_SSTX3-
A13	GBE0_MDI0+	B13	SMB_CK	C13	USB_SSRX3+	D13	USB_SSTX3+
A14	GBE0_CTREF	B14	SMB_DAT	C14	GND	D14	GND
A15	SUS_S3#	B15	SMB_ALERT#	C15	DDI1_PAIR6+	D15	DDI1_CTRLCLK_AUX+
A16	SATA0_TX+	B16	SATA1_TX+	C16	DDI1_PAIR6-	D16	DDI1_CTRLDATA_AUX-
A17	SATA0_TX-	B17	SATA1_TX-	C17	RSVD	D17	RSVD
A18	SUS_S4#	B18	SUS-STAT#	C18	RSVD	D18	RSVD
A19	SATA0_RX+	B19	SATA1_RX+	C19	PCIE_RX6+	D19	PCIE_TX6+
A20	SATA0_RX-	B20	SATA1_RX-	C20	PCIE_RX6-	D20	PCIE_TX6-
A21	GND (FIXED)	B21	GND (FIXED)	C21	GND (FIXED)	D21	GND (FIXED)
A22	SATA2_TX+	B22	SATA3_TX+	C22	PCIE_RX7+	D22	PCIE_TX7+
A23	SATA2_TX-	B23	SATA3_TX-	C23	PCIE_RX7-	D23	PCIE_TX7-
A24	SUS_S5#	B24	PWR_OK	C24	DDI1_HPD	D24	RSVD
A25	SATA2_RX+	B25	SATA3_RX+	C25	DDI1_PAIR4+	D25	RSVD
A26	SATA2_RX-	B26	SATA3_RX-	C26	DDI1_PAIR4-	D26	DDI1_PAIR0+
A27	BATLOW#	B27	WDT	C27	RSVD	D27	DDI1_PAIR0-
A28	SATA_ACT#	B28	HDA_SDIN2	C28	RSVD	D28	RSVD
A29	HDA_SYNC	B29	HDA_SDIN1	C29	DDI1_PAIR5+	D29	DDI1_PAIR1+
A30	HDA_RST#	B30	HDA_SDIN0	C30	DDI1_PAIR5-	D30	DDI1_PAIR1-
A31	GND (FIXED)	B31	GND (FIXED)	C31	GND (FIXED)	D31	GND (FIXED)
A32	HDA_BITCLK	B32	SPKR	C32	DDI2_CTRLCLK_AUX+	D32	DDI1_PAIR2+

Row A		Row B		Row C		Row D	
Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
A33	HDA_SDOUT	B33	I2C_CK	C33	DDI2_CTRLDATA_AUX-	D33	DDI1_PAIR2-
A34	BIOS_DIS0#	B34	I2C_DAT	C34	DDI2_DDC_AUX_SEL	D34	DDI1_DDC_AUX_SEL
A35	THRMTRIP#	B35	THRM#	C35	RSVD	D35	RSVD
A36	USB6-	B36	USB7-	C36	DDI3_CTRLCLK_AUX+	D36	DDI1_PAIR3+
A37	USB6+	B37	USB7+	C37	DDI3_CTRLDATA_AUX-	D37	DDI1_PAIR3-
A38	USB_6_7_OC#	B38	USB_4_5_OC#	C38	DDI3_DDC_AUX_SEL	D38	RSVD
A39	USB4-	B39	USB5-	C39	DDI3_PAIR0+	D39	DDI2_PAIR0+
A40	USB4+	B40	USB5+	C40	DDI3_PAIR0-	D40	DDI2_PAIR0-
A41	GND (FIXED)	B41	GND (FIXED)	C41	GND (FIXED)	D41	GND (FIXED)
A42	USB2-	B42	USB3-	C42	DDI3_PAIR1+	D42	DDI2_PAIR1+
A43	USB2+	B43	USB3+	C43	DDI3_PAIR1-	D43	DDI2_PAIR1-
A44	USB_2_3_OC#	B44	USB_0_1_OC#	C44	DDI3_HPD	D44	DDI2_HPD
A45	USB0-	B45	USB1-	C45	RSVD	D45	RSVD
A46	USB0+	B46	USB1+	C46	DDI3_PAIR2+	D46	DDI2_PAIR2+
A47	VCC_RTC	B47	EXCD1_PERTST#	C47	DDI3_PAIR2-	D47	DDI2_PAIR2-
A48	EXCD0_PERST#	B48	EXCD1_CPPE#	C48	RSVD	D48	RSVD
A49	EXCD0CPPE#	B49	SYS_RESET#	C49	DDI3_PAIR3+	D49	DDI2_PAIR3+
A50	LPC_SERIRQ	B50	CB_RESET#	C50	DDI3_PAIR3-	D50	DDI2_PAIR3-
A51	GND (FIXED)	B51	GND (FIXED)	C51	GND (FIXED)	D51	GND (FIXED)
A52	PCIE_TX5+	B52	PCIE_RX5+	C52	PEG_RX0+	D52	PEG_TX0+
A53	PCIE_TX5-	B53	PCIE_RX5-	C53	PEG_RX0-	D53	PEG_TX0-
A54	GPI0	B54	GPO1	C54	NC	D54	PEG_LANE_RV#
A55	PCIE_TX4+	B55	PCIE_RX4+	C55	PEG_RX1+	D55	PEG_TX1+
A56	PCIE_TX4-	B56	PCIE_RX4-	C56	PEG_RX1-	D56	PEG_TX1-
A57	GND	B57	GPO2	C57	NC	D57	NC
A58	PCIE_TX3+	B58	PCIE_RX3+	C58	PEG_RX2+	D58	PEG_TX2+
A59	PCIE_TX3-	B59	PCIE_RX3-	C59	PEG_RX2-	D59	PEG_TX2-
A60	GND (FIXED)	B60	GND (FIXED)	C60	GND (FIXED)	D60	GND (FIXED)
A61	PCIE_TX2+	B61	PCIE_RX2+	C61	PEG_RX3+	D61	PEG_TX3+
A62	PCIE_TX2-	B62	PCIE_RX2-	C62	PEG_RX3-	D62	PEG_TX3-
A63	GPI1	B63	GPO3	C63	RSVD	D63	RSVD
A64	PCIE_TX1+	B64	PCIE_RX1+	C64	RSVD	D64	RSVD
A65	PCIE_TX1-	B65	PCIE_RX1-	C65	PEG_RX4+	D65	PEG_TX4+
A66	GND	B66	WAKE0#	C66	PEG_RX4-	D66	PEG_TX4-
A67	GPI2	B67	WAKE1#	C67	RSVD	D67	GND
A68	PCIE_TX0+	B68	PCIE_RX0+	C68	PEG_RX5+	D68	PEG_TX5+
A69	PCIE_TX0-	B69	PCIE_RX0-	C69	PEG_RX5-	D69	PEG_TX5-
A70	GND (FIXED)	B70	GND (FIXED)	C70	GND (FIXED)	D70	GND (FIXED)
A71	LVDS_A0+	B71	LVDS_B0+	C71	PEG_RX6+	D71	PEG_TX6+
A72	LVDS_A0-	B72	LVDS_B0-	C72	PEG_RX6-	D72	PEG_TX6-
A73	LVDS_A1+	B73	LVDS_B1+	C73	GND	D73	GND
A74	LVDS_A1-	B74	LVDS_B1-	C74	PEG_RX7+	D74	PEG_TX7+
A75	LVDS_A2+	B75	LVDS_B2+	C75	PEG_RX7-	D75	PEG_TX7-
A76	LVDS_A2-	B76	LVDS_B2-	C76	GND	D76	GND
A77	LVDS_VDD_EN	B77	LVDS_B3+	C77	RSVD	D77	RSVD
A78	LVDS_A3+	B78	LVDS_B3-	C78	PEG_RX8+	D78	PEG_TX8+
A79	LVDS_A3-	B79	LVDS_BKLT_EN	C79	PEG_RX8-	D79	PEG_TX8-

Row A		Row B		Row C		Row D	
Pin	**Signal**	**Pin**	**Signal**	**Pin**	**Signal**	**Pin**	**Signal**
A80	GND (FIXED)	B80	GND (FIXED)	C80	GND (FIXED)	D80	GND (FIXED)
A81	LVDS_A_CK+	B81	LVDS_B_CK+	C81	PEG_RX9+	D81	PEG_TX9+
A82	LVDS_A_CK-	B82	LVDS_B_CK-	C82	PEG_RX9-	D82	PEG_TX9-
A83	LVDS_I2C_CK	B83	LVDS_BKLT_Ctrl	C83	RSVD	D83	RSVD
A84	LVDS_I2C_DAT	B84	VCC_5V_SBY	C84	GND	D84	GND
A85	GPI3	B85	VCC_5V_SBY	C85	PEG_RX10+	D85	PEG_TX10+
A86	**KBD_RSD#**	B86	VCC_5V_SBY	C86	PEG_RX10-	D86	PEG_TX10-
A87	**KBD_A20GATE**	B87	VCC_5V_SBY	C87	GND	D87	GND
A88	PCIE0_CK_REF+	B88	BIOS_DIS1#	C88	PEG_RX11+	D88	PEG_TX11+
A89	PCIE0_CK_REF-	B89		C89	PEG_RX11-	D89	PEG_TX11-
A90	GND (FIXED)	B90		C90	GND (FIXED)	D90	GND (FIXED)
A91	SPI_POWER	B91		C91	PEG_RX12+	D91	PEG_TX12+
A92	SPI_MISO	B92		C92	PEG_RX12-	D92	PEG_TX12-
A93	GPO0	B93		C93	GND	D93	GND
A94	SPI_CLK	B94		C94	PEG_RX13+	D94	PEG_TX13+
A95	SPI_MOSI	B95		C95	PEG_RX13-	D95	PEG_TX13-
A96	TPM_PP	B96		C96	GND	D96	GND
A97	NC	B97	SPI_CS#	C97	RSVD	D97	RSVD
A98	SER0_TX	B98	RSVD	C98	PEG_RX14+	D98	PEG_TX14+
A99	SER0_RX	B99	RSVD	C99	PEG_RX14-	D99	PEG_TX14-
A100	GND (FIXED)	B100	GND (FIXED)	C100	GND (FIXED)	D100	GND (FIXED)
A101	SER1_TX	B101	FAN_PWMOUT	C101	PEG_RX15+	D101	PEG_TX15+
A102	SER1_RX	B102	FAN_TACHIN	C102	PEG_RX15-	D102	PEG_TX15-
A103	LID#	B103	SLEEP#	C103	GND	D103	GND
A104	VCC_12V	B104	VCC_12V	C104	VCC_12V	D104	VCC_12V
A105	VCC_12V	B105	VCC_12V	C105	VCC_12V	D105	VCC_12V
A106	VCC_12V	B106	VCC_12V	C106	VCC_12V	D106	VCC_12V
A107	VCC_12V	B107	VCC_12V	C107	VCC_12V	D107	VCC_12V
A108	VCC_12V	B108	VCC_12V	C108	VCC_12V	D108	VCC_12V
A109	VCC_12V	B109	VCC_12V	C109	VCC_12V	D109	VCC_12V
A110	GND (FIXED)	B110	GND (FIXED)	C110	GND (FIXED)	D110	GND (FIXED)

Chapter 3
Drivers Installation

This chapter introduces installation of the following drivers:

- Intel® Chipset Software Installation Utility
- Graphics Driver
- HD Audio Driver
- LAN Driver
- Intel® Management Engine Drivers Installation

3.1 Introduction

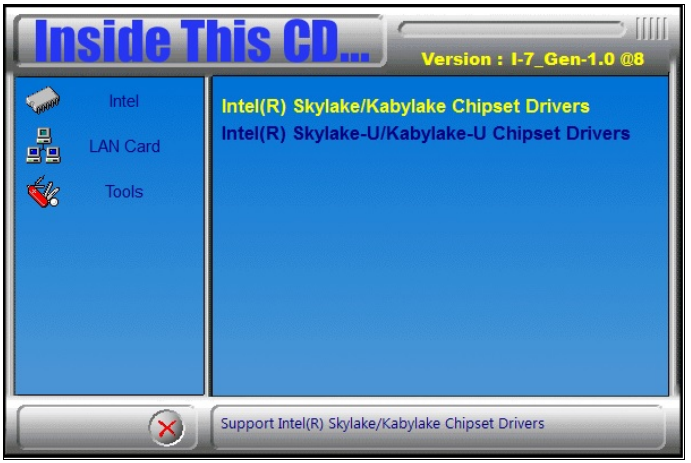
This section describes the installation procedures for software and drivers. The software and drivers are included with the motherboard. If you find anything missing, please contact the distributor where you made the purchase. The contents of this section include the following:

Note: After installing your Windows operating system, you must install the Intel® Chipset Software Installation Utility first before proceeding with the drivers installation.

3.2 Intel® Chipset Software Installation Utility

The Intel® Chipset drivers should be installed first before the software drivers to install INF files for Plug & Play function for Intel chipset components. Follow the instructions below to complete the installation.

1. Insert the disk enclosed in the package with the board. Click **Intel** on the left pane and then **Intel(R) Skylake/Kabylake Chipset Drivers** on the right pane.

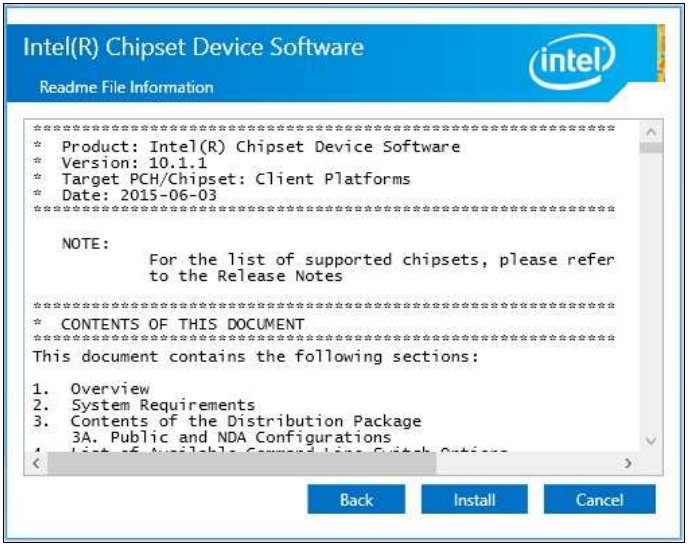


2. Click **Intel(R) Chipset Software Installation Utility**.

3. When the *Welcome* screen to the Intel® Chipset Device Software appears, click **Next** to continue.

4. Accept the software license agreement and proceed with the installation process.

5. On the *Readme File Information* screen, click **Install** for installation.



6. The driver has been completely installed. You are suggested to restart the computer for changes to take effect.

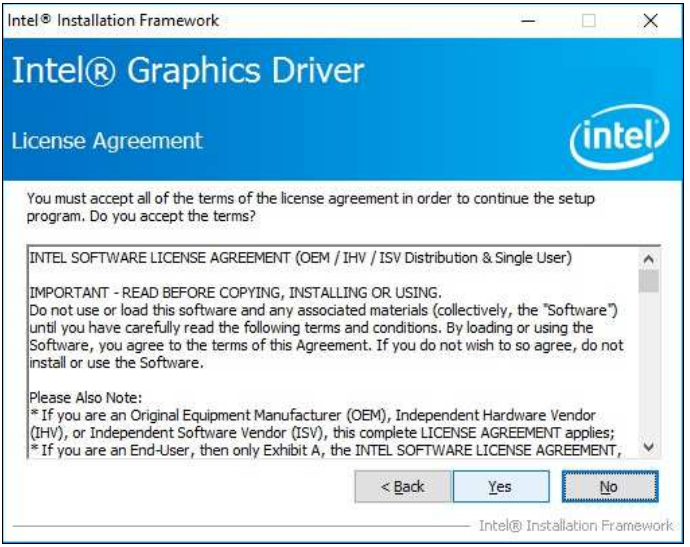
3.3 Graphics Driver Installation

1. Insert the disk enclosed in the package with the board. Click **Intel** on the left pane and then **Intel(R) Skylake/Kabylake Chipset Drivers** on the right pane.

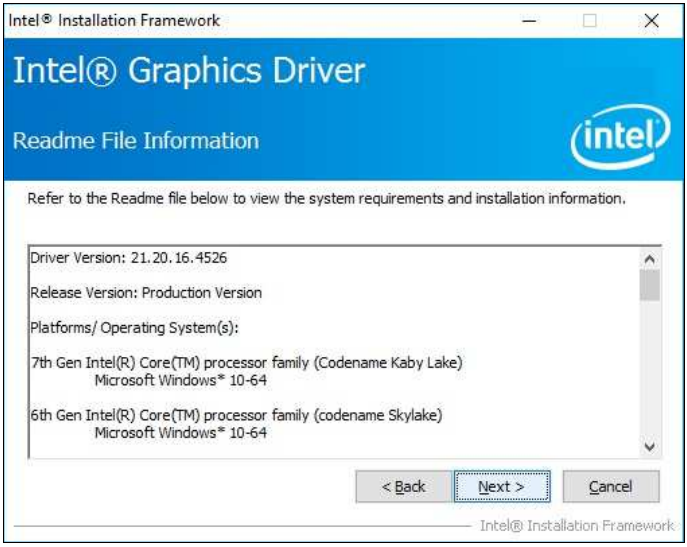
2. Click **Intel(R) HD Graphics Driver**.

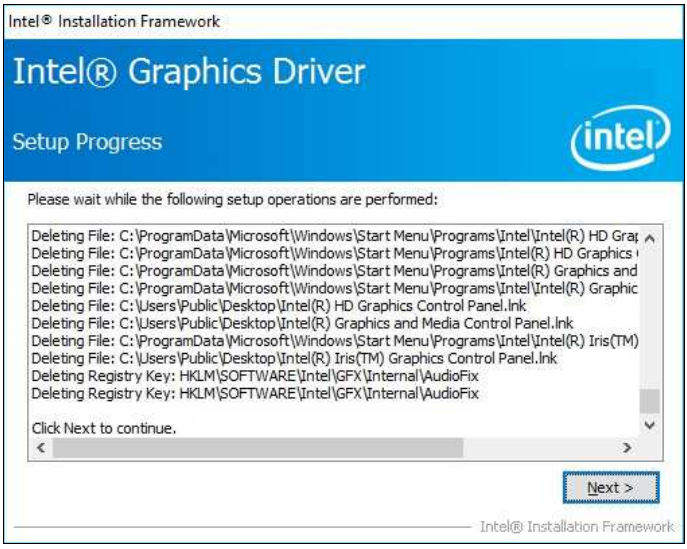
3. When the *Welcome* screen appears, click **Next** to continue.

4. Click **Yes** to accept the license agreement and click **Next** until the installation starts.



5. On the *Readme File Information* screen, click **Next** until the installation starts.

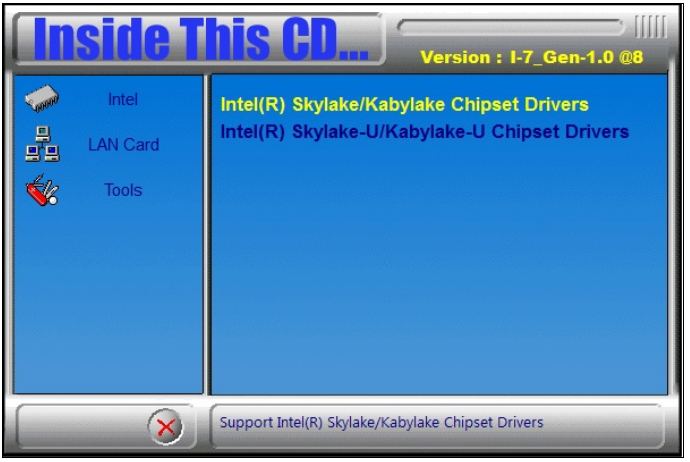




6. The driver has been completely installed. You are suggested to restart the computer for changes to take effect.

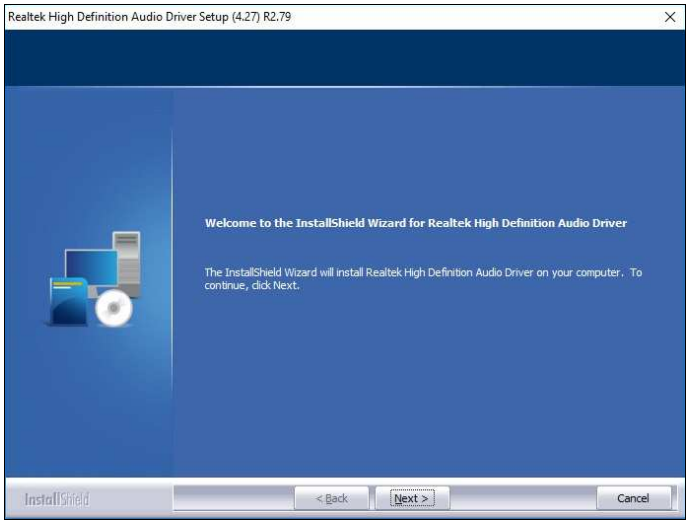
3.4 HD Audio Driver Installation

1. Insert the disk enclosed in the package with the board. Click **Intel** on the left pane and then **Intel(R) Skylake/Kabylake Chipset Drivers** on the right pane.



2. Click **Realtek High Definition Audio Driver**.

3. On the *Welcome* screen of the InstallShield Wizard, click **Next**.

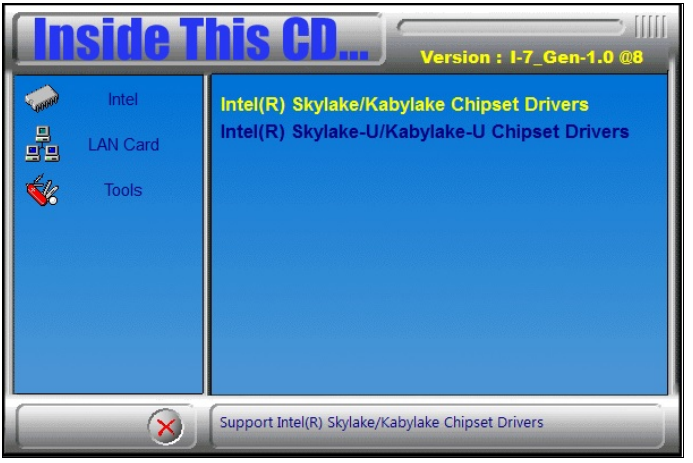


4. Click **Next** until the installation starts.

5. The driver has been completely installed. You are suggested to restart the computer for changes to take effect.

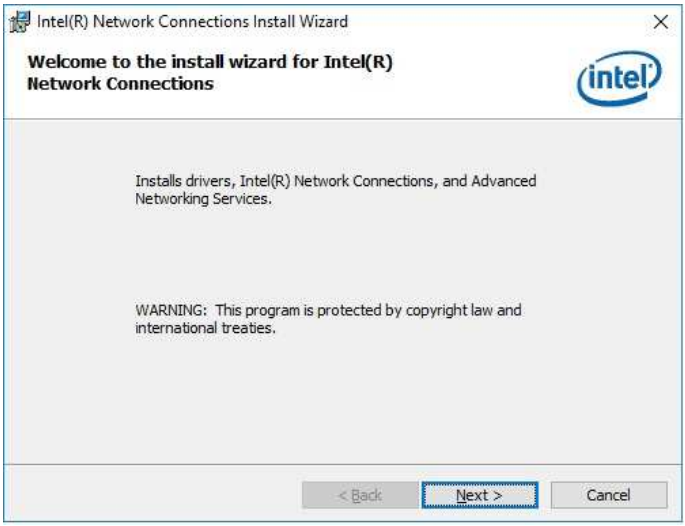
3.5 LAN Driver Installation

1. Insert the disk enclosed in the package with the board. Click **Intel** on the left pane and then **Intel(R) Kabylake-U Chipset Drivers** on the right pane.



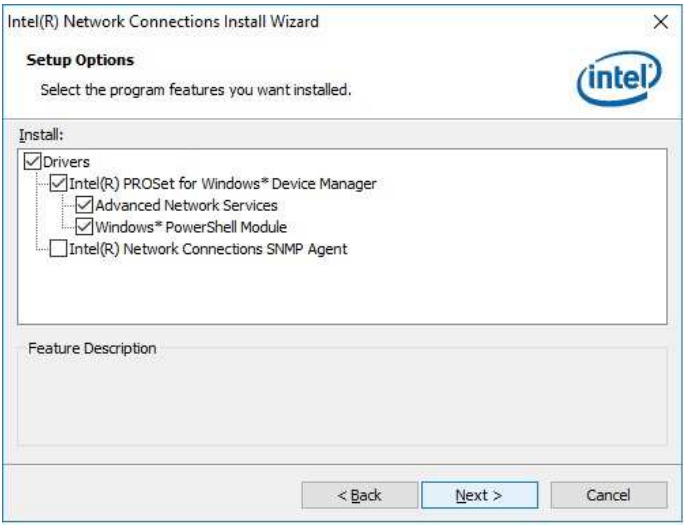
2. Click **Intel(R) PRO LAN Network Drivers**..

3. When the *Welcome* screen appears, click **Next**.

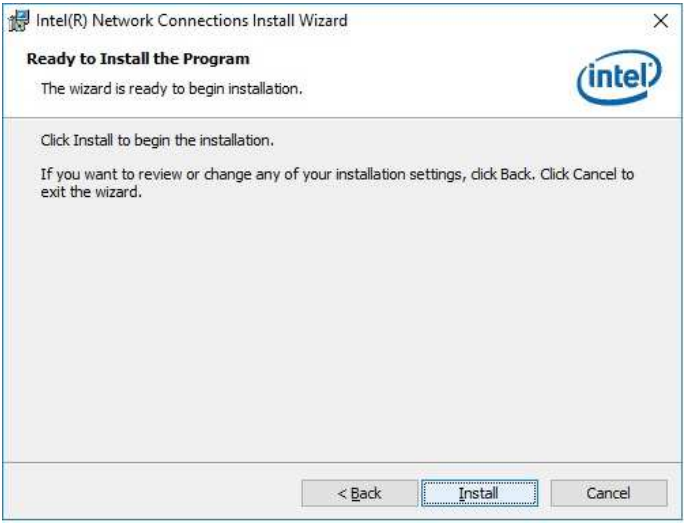


4. Accept the license agreement and click **Next**.

5. On the *Setup Options* screen, click the checkbox to select the desired driver(s) for installation. Then click **Next** to continue.



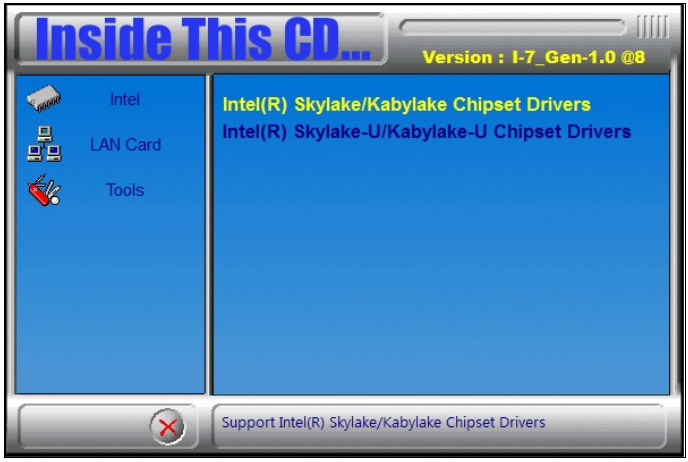
6. The wizard is ready for installation. Click **Install**.



7. As the installation is complete, you are suggested to restart the computer for changes to take effect.

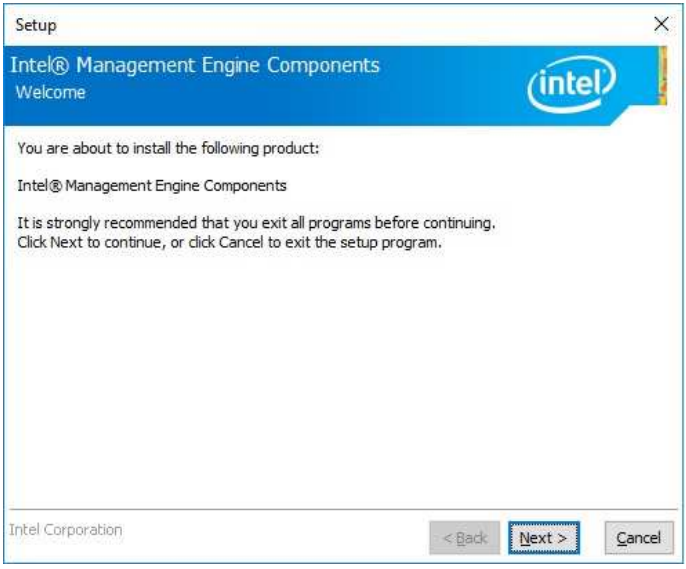
3.6 Intel® Management Engine Drivers Installation

1. Insert the disk enclosed in the package with the board. Click **Intel** on the left pane and then **Intel(R) Skylake/Kabylake Chipset Drivers** on the right pane.

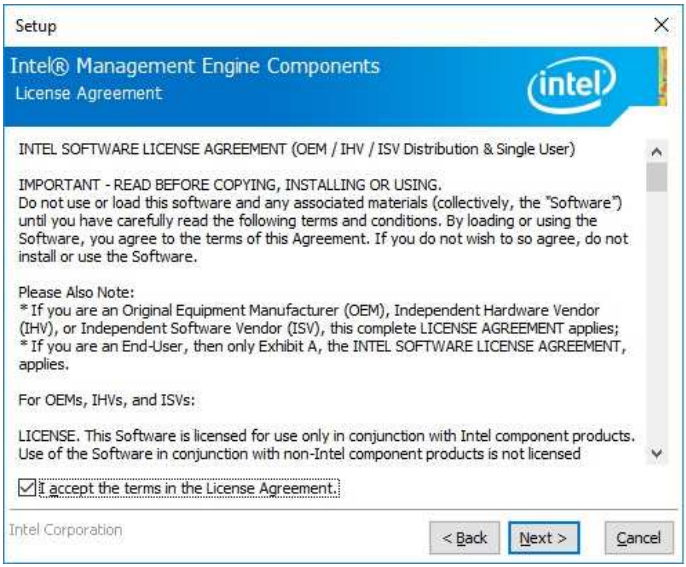


2. Click **Intel(R) ME 11.x Drivers**.

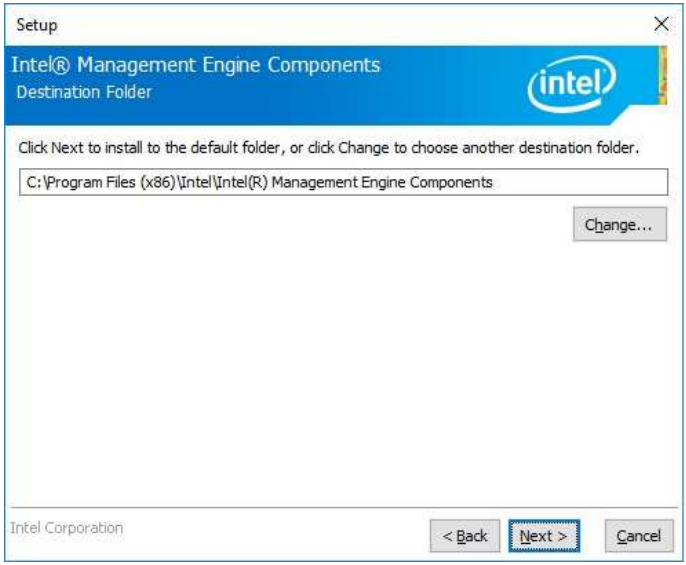
3. When the *Welcome* screen appears, click **Next**.



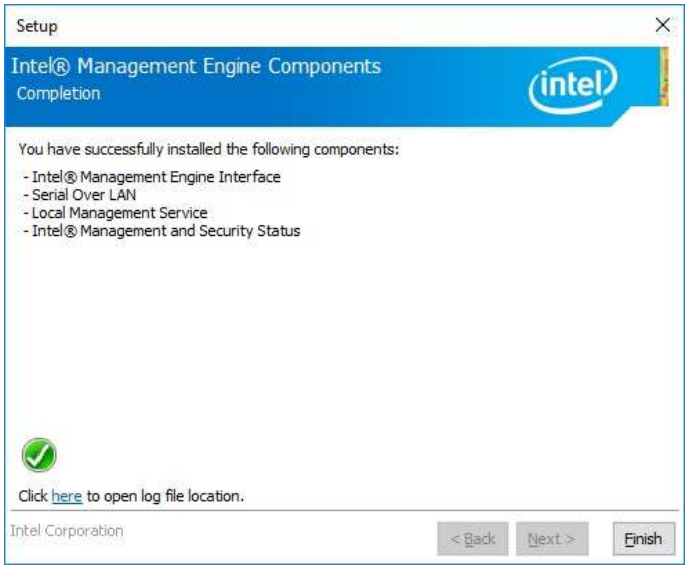
4. Accept the license agreement and click **Next** until the installation starts.



5. Choose a destination folder for installation.



6. As the driver has been sccessfully installed, you are suggested to restart the computer for changes to take effect.



This page is intentionally left blank.

Chapter 4
BIOS Setup

This chapter describes the different settings available in the AMI BIOS that comes with the board. The topics covered in this chapter are as follows:

- Main Settings
- Advanced Settings
- Chipset Settings
- Security Settings
- Boot Settings
- Save & Exit

4.1 Introduction

The BIOS (Basic Input/Output System) installed in the ROM of your computer system supports Intel® processors. The BIOS provides critical low-level support for standard devices such as disk drives, serial ports and parallel ports. It also provides password protection as well as special support for detailed fine-tuning of the chipset controlling the entire system.

4.2 BIOS Setup

The BIOS provides a Setup utility program for specifying the system configurations and settings. The BIOS ROM of the system stores the Setup utility. When you turn on the computer, the BIOS is immediately activated. Press the <Del> key immediately allows you to enter the Setup utility. If you are a little bit late pressing the <Del> key, POST (Power On Self Test) will continue with its test routines, thus preventing you from invoking the Setup.

If you still need to enter Setup, restart the system by pressing the ”Reset” button or simultaneously pressing the <Ctrl>, <Alt> and <Delete> keys. You can also restart by turning the system Off and back On again.

The following message will appear on the screen:

```
Press  <DEL>  to  Enter  Setup
```

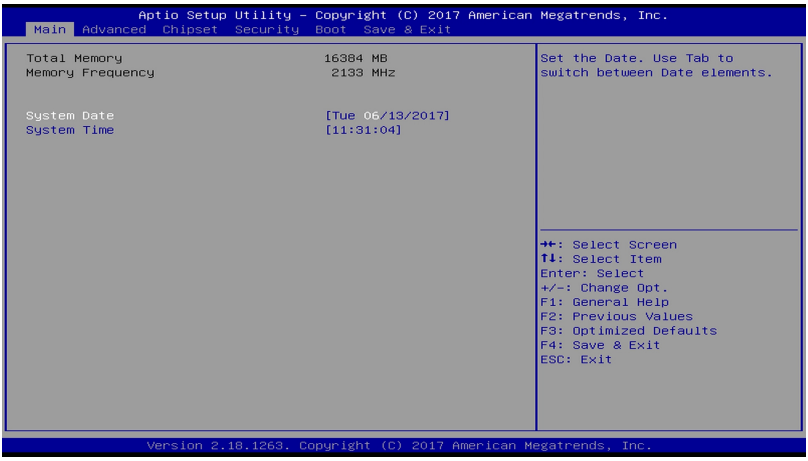
In general, press the arrow keys to highlight items, <Enter> to select, the <PgUp> and <PgDn> keys to change entries, <F1> for help, and <Esc> to quit.

When you enter the BIOS Setup utility, the *Main Menu* screen will appear on the screen. The Main Menu allows you to select from various setup functions and exit choices.

Warning: It is strongly recommended that you avoid making any changes to the chipset defaults.

These defaults have been carefully chosen by both AMI and your system manufacturer to provide the absolute maximum performance and reliability. Changing the defaults could make the system unstable and crash in some cases.

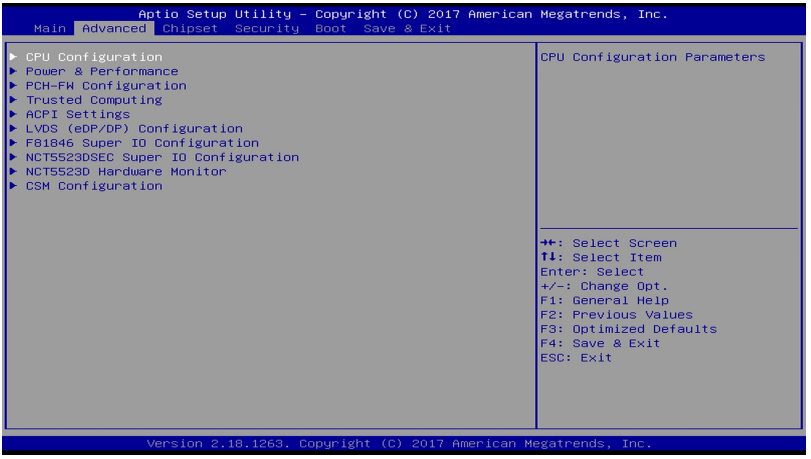
4.3 Main Settings



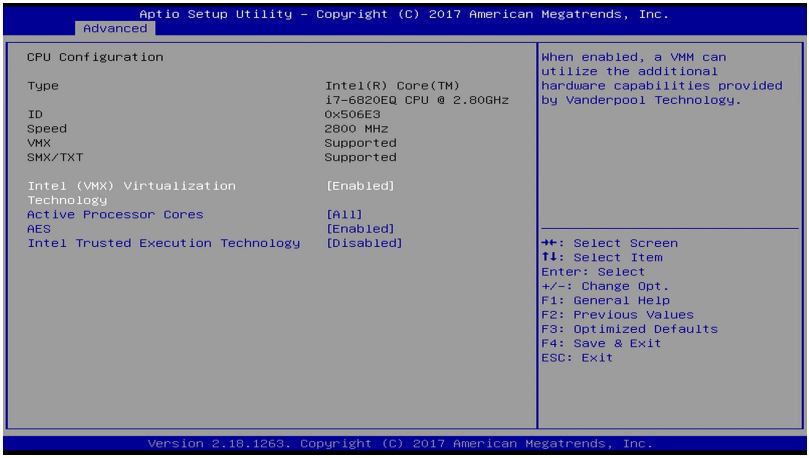
BIOS Setting	Description
System Date	Sets the date. Use the <Tab> key to switch between the data elements.
System Time	Set the time. Use the <Tab> key to switch between the data elements.

4.4 Advanced Settings

This section allows you to configure, improve your system and allows you to set up some system features according to your preference.

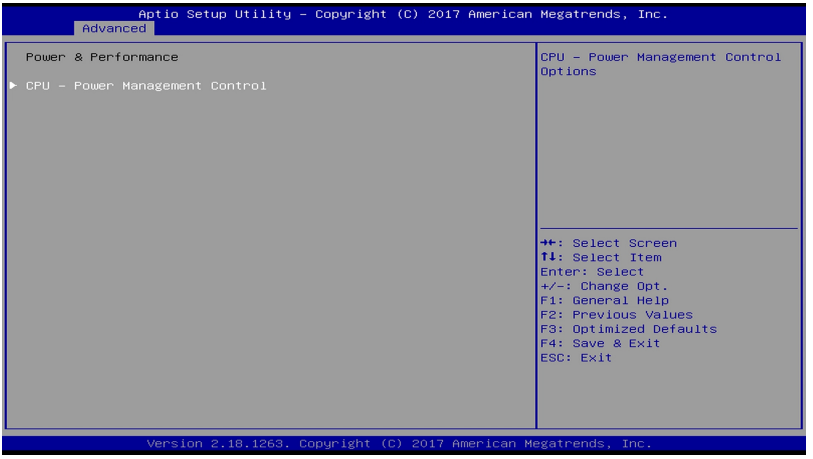


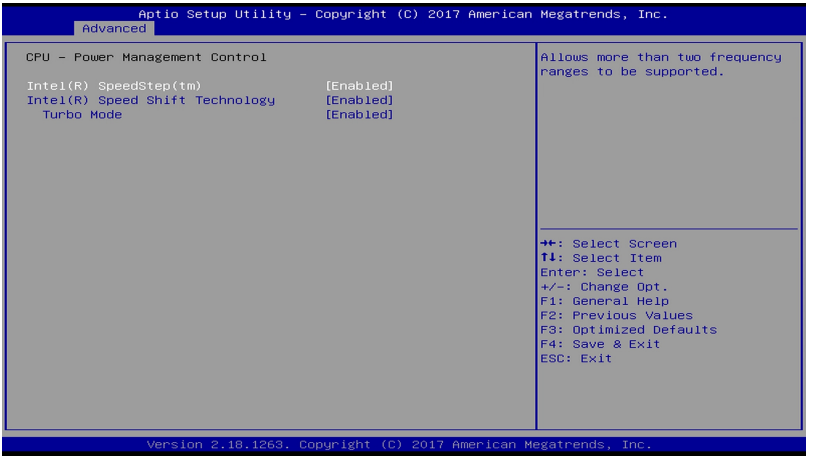
4.4.1 CPU Configuration



BIOS Setting	Description
Intel (VMX) Virtualization Technology	Enables / Disables a VMM can utilize the additional hardware capabilities provided by Vanderpool Technology.
Active Processor Cores	Number of cores to enable in each processor package. Options: All, 1, 2, 3
AES	Enables / Disables AES (Advanced Encryption Standard).
Intel Trusted Execution Technology	Enables / Disables unilization of additional hardware capabilities provided by Intel(R) Trusted Execution Technology. Changes require a full power cycle to take effect.

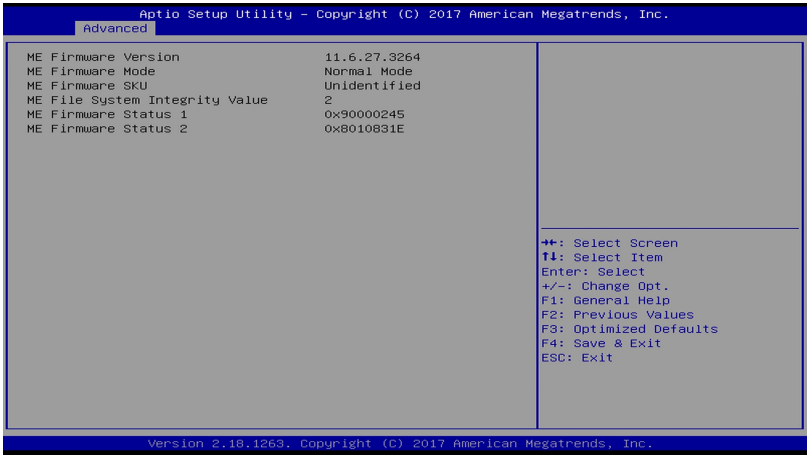
4.4.2 Power & Performance





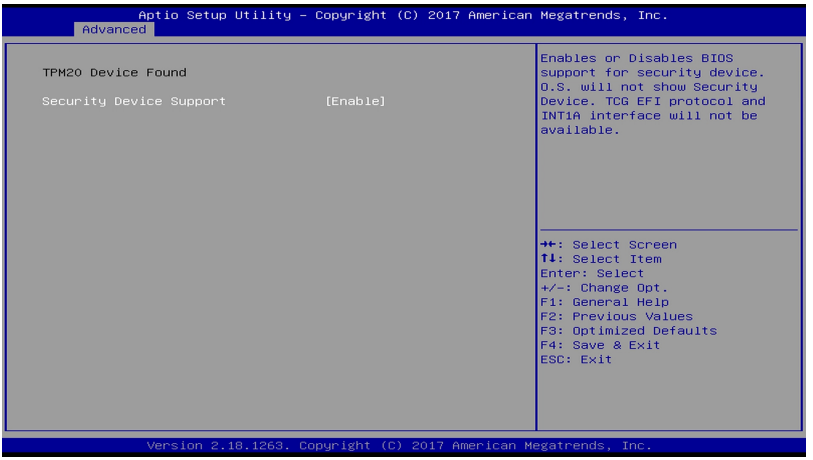
BIOS Setting	Description
Intel(R) SpeedStep(tm)	Allows more than two frequency ranges to be supported.
Intel(R) Speed Shift Technology	Enables / Disables Intel(R) Speed Shift Technology support. Enabling will expose the CPPC v2 interface to allow for hardware controlled P-states.
Turbo Mode	Enables / Disables processor Turbo Mode (requires EMTTM enabled too).

4.4.3 PCH-FW Configuration



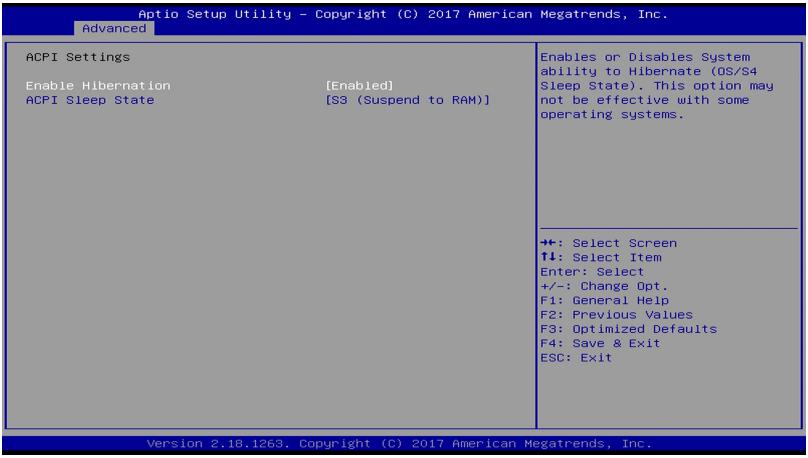
Displays the information of PCH firmware, such as the firmware version, mode, integrity value, and status.

4.4.3.1. Trusted Computing



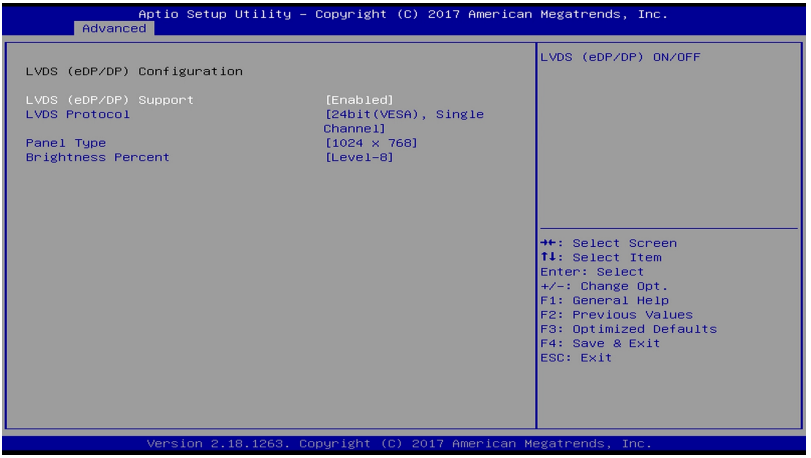
BIOS Setting	Description
Security Device Support	Enables / Disables BIOS support for security device. OS will not show security device. TCG EFI protocol and INT1A interface will not be available.

4.4.4 ACPI Settings



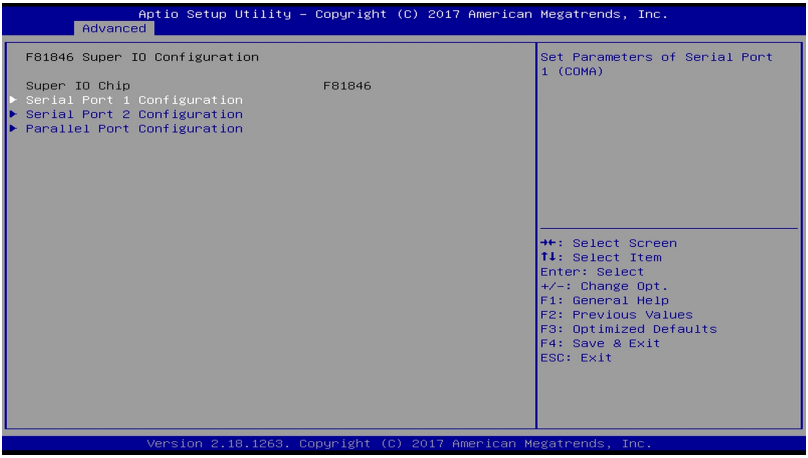
BIOS Setting	Description
Enable Hibernation	Enables / Disables the system ability to hibernate (OS/S4 Sleep State). This option may be not effective with some OS.
ACPI Sleep State	Selects an ACPI sleep state (Suspend Disabled or S3) where the system will enter when the Suspend button is pressed.

4.4.5 LVDS (eDP/DP) Configuration



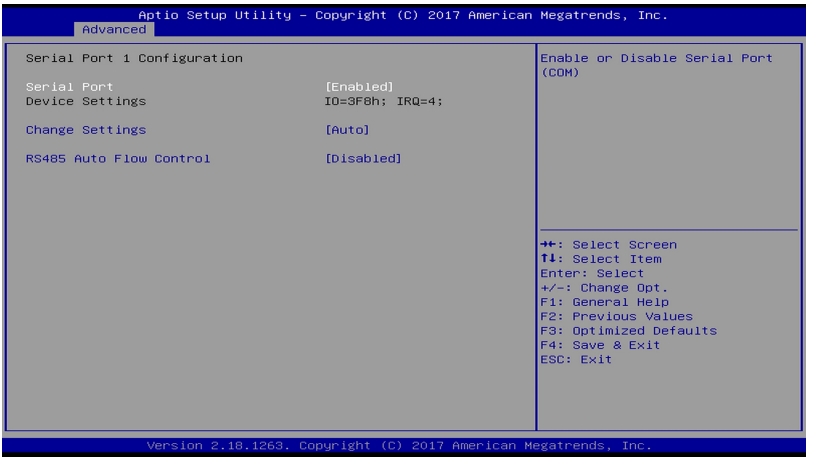
BIOS Setting	Description
LVDS (eDP/DP) Support	Enables / Disables LVDS (eDP/DP).
LVDS Protocol	Selects a LVDS protocol type. Options: 18 bit, Single Channel / 18 bit, Dual Channel / 24 bit (VESA), Single Channel / 24 bit (VESA), Dual Channel / 24 bit (JEIDA), Single Channel / 24 bit (JEIDA) Single Channel / 24 bit (JEIDA), Dual Channel
Panel Type	Selects the resolution of your panle. Options: 800 x 600 / 1024 x 768 / 1280 x 1024 / 1366 x 768 / 1440 x 900 / 1600 x 900 / 1920 x 1080
Brightness Percent	Selects a level of brightness. Options: Level-1 ~ Level-8

4.4.6 F81846 Super IO Configuration



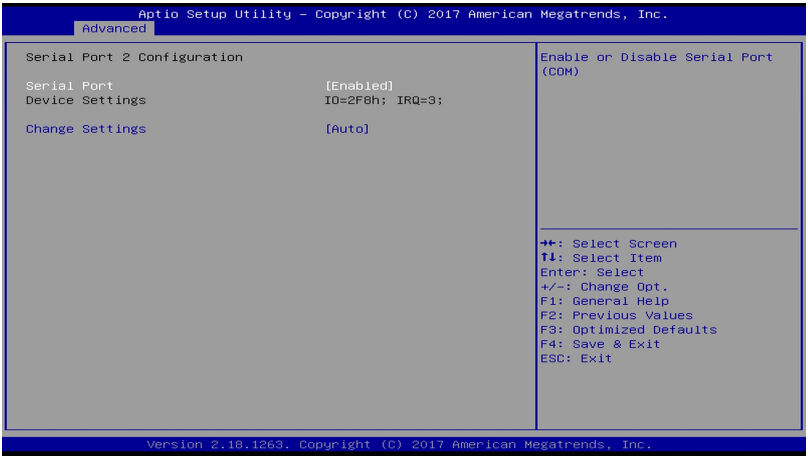
BIOS Setting	Description
Serial Port 1 Configuration	Set parameters of Serial Port 1 (COMA).
Serial Port 2 Configuration	Set parameters of Serial Port 1 (COMB).
Parallel Port Configuration	Set parameters of parallel port (LPT/LPTE).

4.4.6.1. Serial Port 1 Configuration



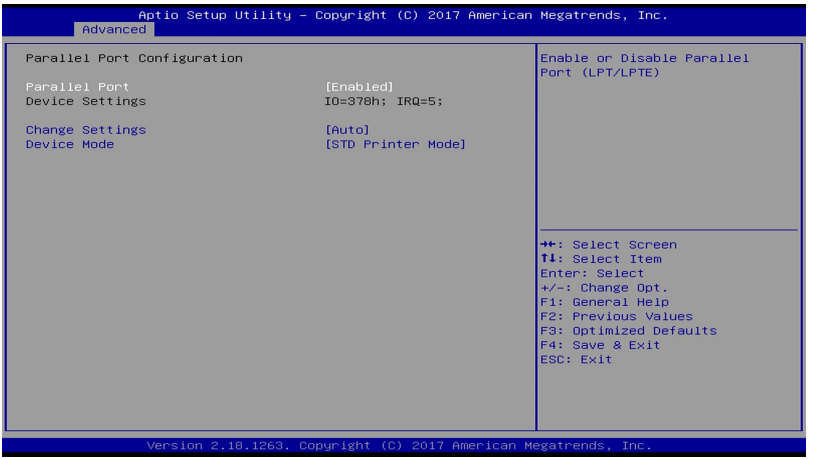
BIOS Setting	Description
Serial Port	Enables / Disables the serial port.
Change Settings	Selects an optimal settings for Super I/O device. Options: • Auto • IO = 3F8h; IRQ = 4 • IO = 3F8h; IRQ = 3, 4, 5, 6, 7, 9, 10, 11, 12 • IO = 2F8h; IRQ = 3, 4, 5, 6, 7, 9, 10, 11, 12 • IO = 3E8h; IRQ = 3, 4, 5, 6, 7, 9, 10, 11, 12 • IO = 2E8h; IRQ = 3, 4, 5, 6, 7, 9, 10, 11, 12
RS485 Auto Flow Control	Enables / Disables RS485 Auto Flow Control.

4.4.6.2. Serial Port 2 Configuration



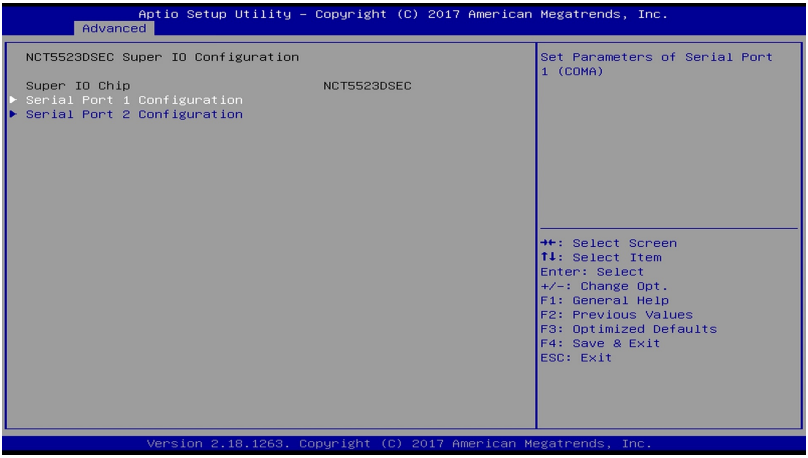
BIOS Setting	Description
Serial Port	Enables / Disables the serial port.
Change Settings	Selects an optimal settings for Super I/O device. Options: • Auto • IO = 2F8h; IRQ = 3 • IO = 3F8h; IRQ = 3, 4, 5, 6, 7, 9, 10, 11, 12 • IO = 2F8h; IRQ = 3, 4, 5, 6, 7, 9, 10, 11, 12 • IO = 3E8h; IRQ = 3, 4, 5, 6, 7, 9, 10, 11, 12 • IO = 2E8h; IRQ = 3, 4, 5, 6, 7, 9, 10, 11, 12

4.4.6.3. Parallel Port Configuration



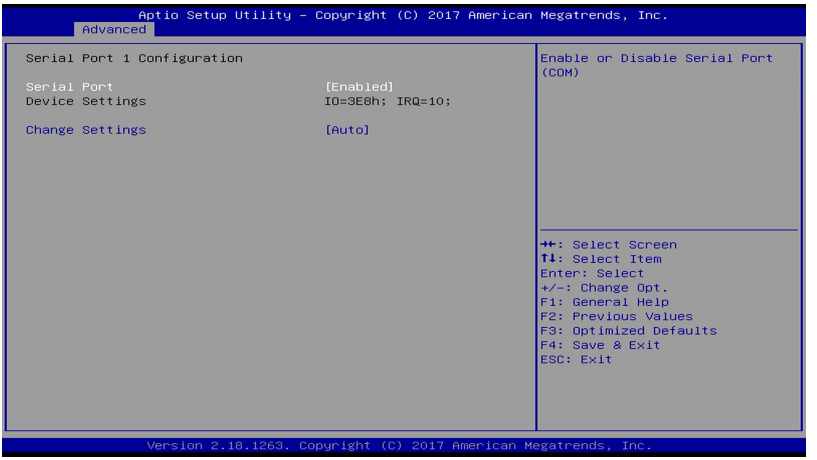
BIOS Setting	Description
Parallel Port	Enables / Disables parallel port (LPT/LPTE)
Change Settings	Selects an optimal settings for Super I/O device. Options: • Auto • IO = 378h; IRQ = 5 • IO = 378h; IRQ = 5, 6, 7, 9, 10, 11, 12 • IO = 278h; IRQ = 5, 6, 7, 9, 10, 11, 12 • IO = 3BCh; IRQ =5, 6, 7, 9, 10, 11, 12
Device Mode	Changes the printer port mode. Options: STD Printer Mode / SPP Mode / EPP-1.9 and SPP Mode / EPP-1.7 and SPP Mode / ECP Mode / ECP and EPP 1.9 Mode / ECP and EPP 1.7 Mode

4.4.7 NCT5523DSEC Super IO Configuration



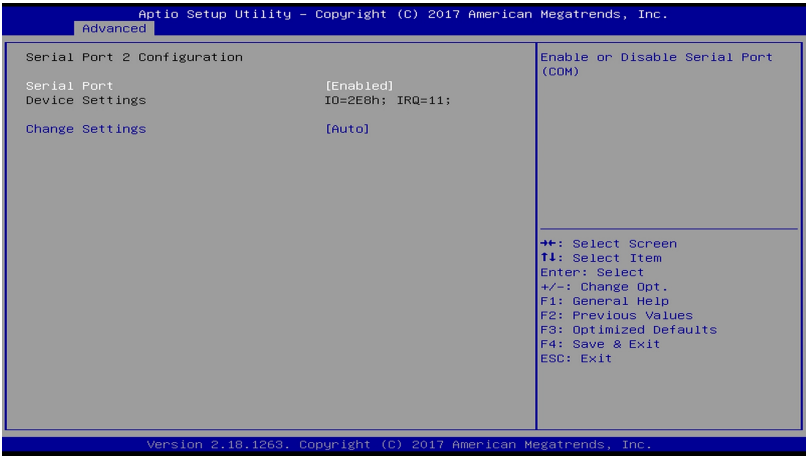
BIOS Setting	Description
Serial Port 1 Configuration	Set parameters of Serial Port 1 (COMA).
Serial Port 2 Configuration	Set parameters of Serial Port 1 (COMB).

4.4.7.1. Serial Port 1 Configuration



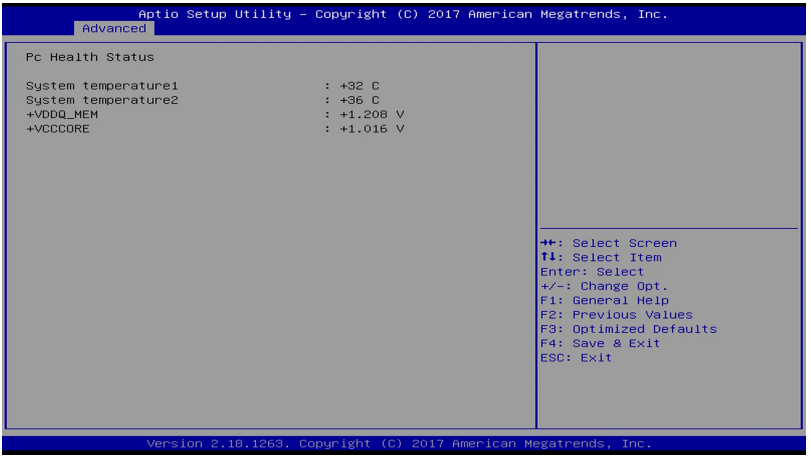
BIOS Setting	Description
Serial Port	Enables / Disables the serial port.
Change Settings	Selects an optimal settings for Super I/O device. Options: • Auto • IO = 3E8h; IRQ = 10 • IO = 240h; IRQ = 3, 4, 5, 6, 7, 10, 11, 12 • IO = 248h; IRQ = 3, 4, 5, 6, 7, 10, 11, 12 • IO = 250h; IRQ = 3, 4, 5, 6, 7, 10, 11, 12 • IO = 258h; IRQ = 3, 4, 5, 6, 7, 10, 11, 12 • IO = 260h; IRQ = 3, 4, 5, 6, 7, 10, 11, 12 • IO = 268h; IRQ = 3, 4, 5, 6, 7, 10, 11, 12

4.4.7.2. Serial Port 2 Configuration



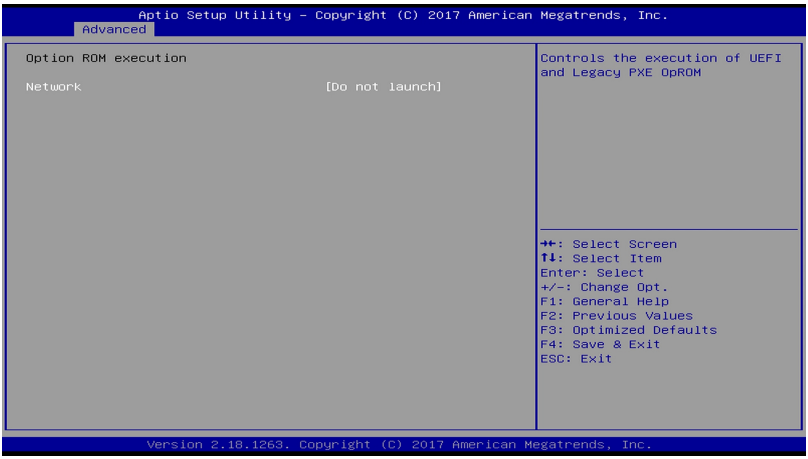
BIOS Setting	Description
Serial Port	Enables / Disables the serial port.
Change Settings	Selects an optimal settings for Super I/O device. Options: • Auto • IO = 2E8h; IRQ = 11 • IO = 240h; IRQ = 3, 4, 5, 6, 7, 10, 11, 12 • IO = 248h; IRQ = 3, 4, 5, 6, 7, 10, 11, 12 • IO = 250h; IRQ = 3, 4, 5, 6, 7, 10, 11, 12 • IO = 258h; IRQ = 3, 4, 5, 6, 7, 10, 11, 12 • IO = 260h; IRQ = 3, 4, 5, 6, 7, 10, 11, 12 • IO = 268h; IRQ = 3, 4, 5, 6, 7, 10, 11, 12

4.4.8 NCT5523D Hardware Monitor



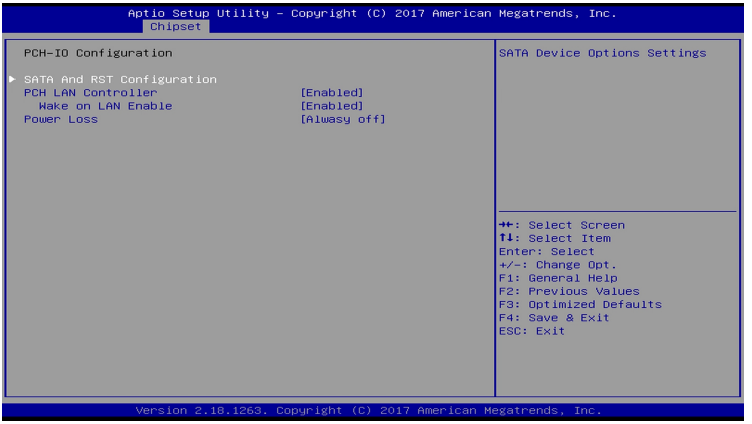
Displays the information of the computer health status.

4.4.9 CSM Configuration



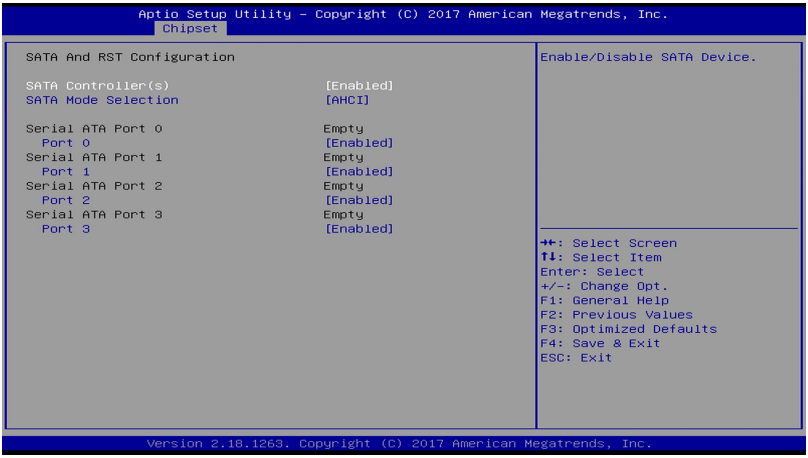
BIOS Setting	Description
Network	Controls the execution of UEFI and Legacy PXE OpROM. Options: Do not launcy / Legacy

4.5 Chipset Settings



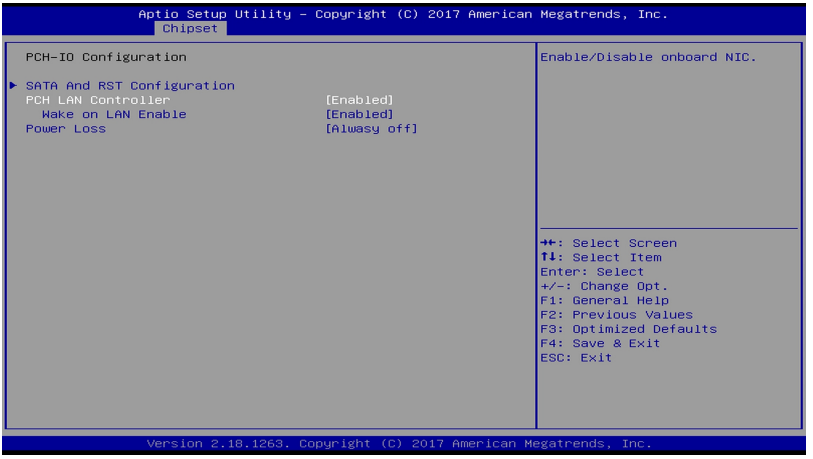
BIOS Setting	Description
SATA and RST Configuration	SATA device options settings.
PCH LAN Controller	Enables / Disables onboard NIC.
Wake on LAN Enable	Enables / Disables integrated LAN to wake the system.
Power Loss	Specify the state to go to when power is re-applied after a power failure (G3 state). Options:Always on / Always off

4.5.1 SATA and RST Configuration



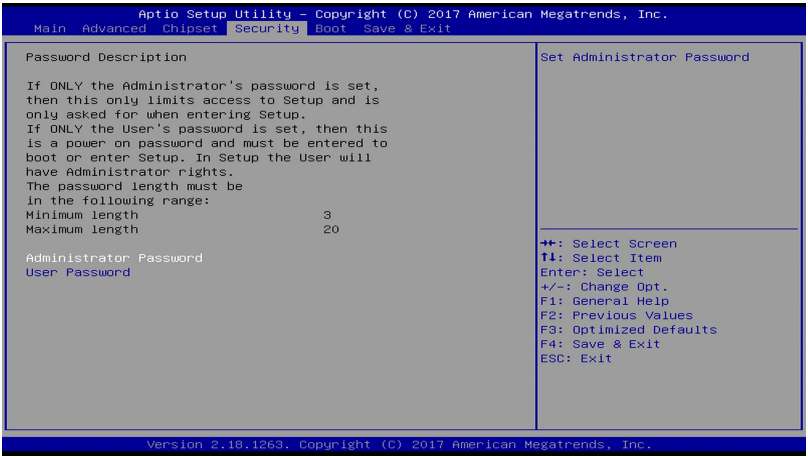
BIOS Setting	Description
SATA Controller(s)	Enables / Disables the Serial ATA.
SATA Mode Selection	Selects IDE or AHCI Mode.
Serial ATA Port 0~2	Enables / Disables Serial Port 0 ~ 2.

4.5.2 PCH-IO Configuration



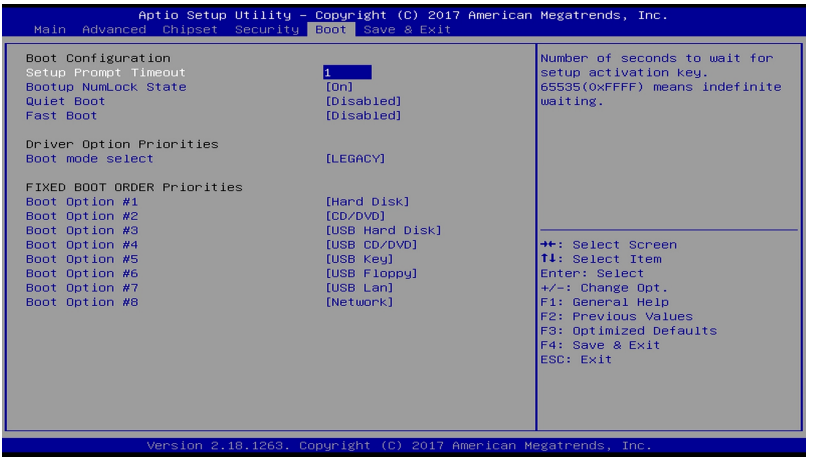
BIOS Setting	Description
SATA and RST Configuration	SATA device options and settings
PCH LAN Controller	Enables / Disables onboard NIC.
Wake on LAN Enable	Enables / Disables integrated LAN to wake the system.
Power Loss	Specify what state to go to when power is re-applied after a power failure (G3 state). Options: Always on / Always off

4.6 Security Settings



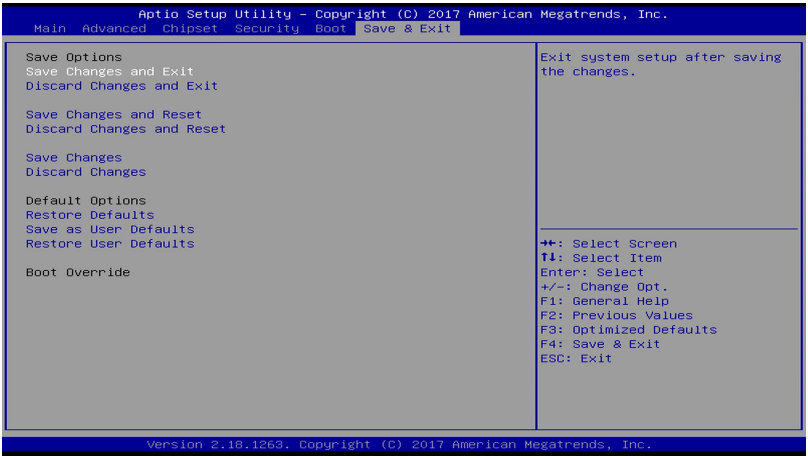
BIOS Setting	Description
Setup Administrator Password	Sets an administrator password for the setup utility.
User Password	Sets a user password.

4.7 Boot Settings



BIOS Setting	Description
Setup Prompt Timeout	Number of seconds to wait for setup activation key. 65535 (0xFFFF) means indefinite waiting.
Bootup NumLock State	Selects the keyboard NumLock state.
Quiet Boot	Enables / Disables Quiet Boot option.
Fast Boot	Enables / Disables boot with initialization of a minimal set of devices required to launch the active boot option. Has no effect for BBS boot options.
Boot mode select	Selects a Boot mode, Legacy / UEFI / Dual.
Boot Option Priorities	Sets the system boot order priorities for hard disk, CD/DVD, USB, Network.

4.8 Save & Exit Settings



BIOS Setting	Description
Save Changes and Exit	Exits system setup after saving the changes.
Discard Changes and Exit	Exits system setup without saving any changes.
Save Changes and Reset	Resets the system after saving the changes.
Discard Changes and Reset	Resets system setup without saving any changes.
Save Changes	Saves changes done so far to any of the setup options.
Discard Changes	Discards changes done so far to any of the setup options.
Restore Defaults	Restores / Loads defaults values for all the setup options.
Save as User Defaults	Saves the changes done so far as User Defaults.
Restore User Defaults	Restores the user defaults to all the setup options.

This page is intentionally left blank.

Appendix

This section provides the mapping addresses of peripheral devices and the sample code of watchdog timer configuration.

A. I/O Port Address Map

Each peripheral device in the system is assigned a set of I/O port addresses which also becomes the identity of the device. The following table lists the I/O port addresses used.

Address	Device Description
0x0000F000-0x0000F03F	Intel(R) HD Graphics P630
0x000003B0-0x000003BB	Intel(R) HD Graphics P630
0x000003C0-0x000003DF	Intel(R) HD Graphics P630
0x00000A00-0x00000A0F	Motherboard resources
0x00000A10-0x00000A1F	Motherboard resources
0x00000A20-0x00000A2F	Motherboard resources
0x0000002E-0x0000002F	Motherboard resources
0x0000004E-0x0000004F	Motherboard resources
0x00000061-0x00000061	Motherboard resources
0x00000063-0x00000063	Motherboard resources
0x00000065-0x00000065	Motherboard resources
0x00000067-0x00000067	Motherboard resources
0x00000070-0x00000070	Motherboard resources
0x00000070-0x00000070	System CMOS/real time clock
0x00000080-0x00000080	Motherboard resources
0x00000092-0x00000092	Motherboard resources
0x000000B2-0x000000B3	Motherboard resources
0x00000680-0x0000069F	Motherboard resources
0x0000FFFF-0x0000FFFF	Motherboard resources
0x0000FFFF-0x0000FFFF	Motherboard resources
0x0000FFFF-0x0000FFFF	Motherboard resources
0x00001800-0x000018FE	Motherboard resources
0x0000164E-0x0000164F	Motherboard resources
0x00000378-0x0000037F	Printer Port (LPT1)

Address	Device Description
0x00000A30-0x00000A3F	Motherboard resources
0x00000040-0x00000043	System timer
0x00000050-0x00000053	System timer
0x00000800-0x0000087F	Motherboard resources
0x0000F090-0x0000F097	Standard SATA AHCI Controller
0x0000F080-0x0000F083	Standard SATA AHCI Controller
0x0000F060-0x0000F07F	Standard SATA AHCI Controller
0x000003F8-0x000003FF	Communications Port (COM1)
0x000002F8-0x000002FF	Communications Port (COM2)
0x000003E8-0x000003EF	Communications Port (COM3)
0x000002E8-0x000002EF	Communications Port (COM4)
0x00000000-0x00000CF7	PCI Express Root Complex
0x00000D00-0x0000FFFF	PCI Express Root Complex
0x00000020-0x00000021	Programmable interrupt controller
0x00000024-0x00000025	Programmable interrupt controller
0x00000028-0x00000029	Programmable interrupt controller
0x0000002C-0x0000002D	Programmable interrupt controller
0x00000030-0x00000031	Programmable interrupt controller
0x00000034-0x00000035	Programmable interrupt controller
0x00000038-0x00000039	Programmable interrupt controller
0x0000003C-0x0000003D	Programmable interrupt controller
0x000000A0-0x000000A1	Programmable interrupt controller
0x000000A4-0x000000A5	Programmable interrupt controller
0x000000A8-0x000000A9	Programmable interrupt controller
0x000000AC-0x000000AD	Programmable interrupt controller
0x000000B0-0x000000B1	Programmable interrupt controller
0x000000B4-0x000000B5	Programmable interrupt controller
0x000000B8-0x000000B9	Programmable interrupt controller

Address	Device Description
0x000000BC-0x000000BD	Programmable interrupt controller
0x000004D0-0x000004D1	Programmable interrupt controller
0x0000F0A0-0x0000F0A7	Intel(R) Active Management Technology - SOL (COM5)
0x00001854-0x00001857	Motherboard resources
0x0000FF00-0x0000FFFE	Motherboard resources
0x0000F040-0x0000F05F	Intel(R) 100 Series/C230 Series Chipset Family SMBus - A123
0x00000060-0x00000060	Standard PS/2 Keyboard
0x00000064-0x00000064	Standard PS/2 Keyboard
0x000000F0-0x000000F0	Numeric data processor

B. Interrupt Request Lines (IRQ)

Peripheral devices use interrupt request lines to notify CPU for the service required. The following table shows the IRQ used by the devices on board.

Level	Function
IRQ 0	System timer
IRQ 1	Standard PS/2 Keyboard
IRQ 3	Communications Port (COM2)
IRQ 4	Communications Port (COM1)
IRQ 8	System CMOS/real time clock
IRQ 10	Communications Port (COM3)
IRQ 11	Communications Port (COM4)
IRQ 11	Intel(R) 100 Series/C230 Series Chipset Family SMBus - A123
IRQ 11	Intel(R) 100 Series/C230 Series Chipset Family Thermal subsystem - A131
IRQ 12	Microsoft PS/2 Mouse
IRQ 13	Numeric data processor
IRQ 14	Motherboard resources
IRQ 16	High Definition Audio Controller
IRQ 19	Intel(R) Active Management Technology - SOL (COM5)
IRQ 54 ~ IRQ 204	Microsoft ACPI-Compliant System
IRQ 256 ~ IRQ 511	Microsoft ACPI-Compliant System
IRQ 4294967290	Intel(R) HD Graphics P630
IRQ 4294967291	Intel(R) Management Engine Interface
IRQ 4294967292	Intel(R) USB 3.0 eXtensible Host Controller - 1.0 (Microsoft)
IRQ 4294967293	Intel(R) Ethernet Connection (2) I219-LM
IRQ 4294967294	Standard SATA AHCI Controller

C. Digital I/O Sample Code

1. The file NCT5523D.H

```
//---------------------------------------------------------------------------
// THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY
// KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
// IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR
// PURPOSE.
//---------------------------------------------------------------------------
#ifndef __NCT5523D_H
#define __NCT5523D_H                                    1
//---------------------------------------------------------------------------
#define         NCT5523D_INDEX_PORT             (NCT5523D_BASE)
#define         NCT5523D_DATA_PORT              (NCT5523D_BASE+1)
//---------------------------------------------------------------------------
#define         NCT5523D_REG_LD                         0x07
//---------------------------------------------------------------------------
#define NCT5523D_UNLOCK                                 0x87
#define         NCT5523D_LOCK                           0xAA
//---------------------------------------------------------------------------
unsigned int Init_NCT5523D(void);
void Set_NCT5523D_LD( unsigned char);
void Set_NCT5523D_Reg( unsigned char, unsigned char);
unsigned char Get_NCT5523D_Reg( unsigned char);
//---------------------------------------------------------------------------
#endif          //__NCT5523D_H
```

2. The file MAIN.CPP

```
//---------------------------------------------------------------------------
// THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY
// KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
// IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR
// PURPOSE.
//---------------------------------------------------------------------------
#include <dos.h>
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#include "NCT5523D.H"
//---------------------------------------------------------------------------
int main (void);

void Dio5Initial(void);
void Dio5SetOutput(unsigned char);
unsigned char Dio5GetInput(void);
void Dio5SetDirection(unsigned char);
unsigned char Dio5GetDirection(void);
//---------------------------------------------------------------------------
int main (void)
{
        char SIO;

        SIO = Init_NCT5523D();
        if (SIO == 0)
        {
                printf("Can not detect Nuvoton NCT5523D, program abort.\n");
                return(1);
        }

        Dio5Initial();

        //for GPIO20..27
        Dio5SetDirection(0x0F);         //GP20..23 = input, GP24..27=output
        printf("Current DIO direction = 0x%X\n", Dio5GetDirection());

        printf("Current DIO status = 0x%X\n", Dio5GetInput());

        printf("Set DIO output to high\n");
        Dio5SetOutput(0x0F);

        printf("Set DIO output to low\n");
        Dio5SetOutput(0x00);

        return 0;
}
//---------------------------------------------------------------------------
```

```
void Dio5Initial(void)
{
        unsigned char ucBuf;

   ucBuf = Get_NCT5523D_Reg(0x1C);
   ucBuf &= ~0x02;
   Set_NCT5523D_Reg(0x1C, ucBuf);

        Set_NCT5523D_LD(0x07);
        //switch to logic device 7
        //enable the GP2 group
        ucBuf = Get_NCT5523D_Reg(0x30);
        ucBuf |= 0x04;
        Set_NCT5523D_Reg(0x30, ucBuf);
}
//---------------------------------------------------------------------------
void Dio5SetOutput(unsigned char NewData)
{
        Set_NCT5523D_LD(0x07);
        //switch to logic device 7
        Set_NCT5523D_Reg(0xE1, NewData);
}
//---------------------------------------------------------------------------
unsigned char Dio5GetInput(void)
{
        unsigned char result;

        Set_NCT5523D_LD(0x07);
        //switch to logic device 7
        result = Get_NCT5523D_Reg(0xE1);
        return (result);
}
//---------------------------------------------------------------------------
void Dio5SetDirection(unsigned char NewData)
{
        //NewData : 1 for input, 0 for output
        Set_NCT5523D_LD(0x07);
        //switch to logic device 7
        Set_NCT5523D_Reg(0xE8, NewData);
}
//---------------------------------------------------------------------------
unsigned char Dio5GetDirection(void)
{
        unsigned char result;

        Set_NCT5523D_LD(0x07);
        //switch to logic device 7
        result = Get_NCT5523D_Reg(0xE8);
        return (result);
}
//---------------------------------------------------------------------------
```

3. The file NCT5523D.CPP

```
//---------------------------------------------------------------------------
// THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY
// KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
// IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR
// PURPOSE.
//---------------------------------------------------------------------------
#include "NCT5523D.H"
#include <dos.h>
//---------------------------------------------------------------------------
unsigned int NCT5523D_BASE;
void Unlock_NCT5523D (void);
void Lock_NCT5523D (void);
//---------------------------------------------------------------------------
unsigned int Init_NCT5523D(void)
{
        unsigned int result;
        unsigned char ucDid;

        NCT5523D_BASE = 0x4E;
        result = NCT5523D_BASE;

        ucDid = Get_NCT5523D_Reg(0x20);
        if (ucDid == 0xC4)
        //NCT5523D??
        {       goto Init_Finish;       }

        NCT5523D_BASE = 0x2E;
        result = NCT5523D_BASE;

        ucDid = Get_NCT5523D_Reg(0x20);
        if (ucDid == 0xC4)
        //NCT5523D??
        {       goto Init_Finish;       }

        NCT5523D_BASE = 0x00;
        result = NCT5523D_BASE;

Init_Finish:
        return (result);
}
//---------------------------------------------------------------------------
void Unlock_NCT5523D (void)
{
        outportb(NCT5523D_INDEX_PORT, NCT5523D_UNLOCK);
        outportb(NCT5523D_INDEX_PORT, NCT5523D_UNLOCK);
}
//---------------------------------------------------------------------------
void Lock_NCT5523D (void)
{
        outportb(NCT5523D_INDEX_PORT, NCT5523D_LOCK);
}
//---------------------------------------------------------------------------
```

```
void Set_NCT5523D_LD( unsigned char LD)
{
        Unlock_NCT5523D();
        outportb(NCT5523D_INDEX_PORT, NCT5523D_REG_LD);
        outportb(NCT5523D_DATA_PORT, LD);
        Lock_NCT5523D();
}
//---------------------------------------------------------------------------
void Set_NCT5523D_Reg( unsigned char REG, unsigned char DATA)
{
        Unlock_NCT5523D();
        outportb(NCT5523D_INDEX_PORT, REG);
        outportb(NCT5523D_DATA_PORT, DATA);
        Lock_NCT5523D();
}
//---------------------------------------------------------------------------
unsigned char Get_NCT5523D_Reg(unsigned char REG)
{
        unsigned char Result;
        Unlock_NCT5523D();
        outportb(NCT5523D_INDEX_PORT, REG);
        Result = inportb(NCT5523D_DATA_PORT);
        Lock_NCT5523D();
        return Result;
}
//---------------------------------------------------------------------------
```

D. Watchdog Timer Configuration

The Watchdog Timer (WDT) is used to generate a variety of output signals after a user programmable count. The WDT is suitable for use in the prevention of system lock-up, such as when software becomes trapped in a deadlock. Under these sorts of circumstances, the timer will count to zero and the selected outputs will be driven.

Under normal circumstance, you will need to restart the WDT at regular intervals before the timer counts to zero.

1. Sample Code: The file NCT5523D.H

```
//---------------------------------------------------------------------------
//
// THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY
// KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
// IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR
// PURPOSE.
//
//---------------------------------------------------------------------------
#ifndef __NCT5523D_H
#define __NCT5523D_H                                    1
//---------------------------------------------------------------------------
#define         NCT5523D_INDEX_PORT             (NCT5523D_BASE)
#define         NCT5523D_DATA_PORT              (NCT5523D_BASE+1)
//---------------------------------------------------------------------------
#define         NCT5523D_REG_LD                         0x07
//---------------------------------------------------------------------------
#define NCT5523D_UNLOCK                                 0x87
#define         NCT5523D_LOCK                           0xAA
//---------------------------------------------------------------------------
unsigned int Init_NCT5523D(void);
void Set_NCT5523D_LD( unsigned char);
void Set_NCT5523D_Reg( unsigned char, unsigned char);
unsigned char Get_NCT5523D_Reg( unsigned char);
//---------------------------------------------------------------------------
#endif          //__NCT5523D_H
```

2. Sample Code: The file MAIN.CPP

```
//---------------------------------------------------------------------------
//
// THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY
// KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
// IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR
// PURPOSE.
//
//---------------------------------------------------------------------------
#include <dos.h>
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#include "NCT5523D.H"
//---------------------------------------------------------------------------
int main (void);

void WDTInitial(void);
void WDTEnable(unsigned char);
void WDTDisable(void);

//---------------------------------------------------------------------------
int main (void)
{
        char SIO;

        SIO = Init_NCT5523D();
        if (SIO == 0)
        {
                printf("Can not detect Nuvoton NCT5523D, program abort.\n");
                return(1);
        }

        WDTInitial();

        WDTEnable(10);

        WDTDisable();

        return 0;
}
//---------------------------------------------------------------------------
void WDTInitial(void)
{
        unsigned char bBuf;
        Set_NCT5523D_LD(0x08);                                  //switch
to logic device 8
        bBuf = Get_NCT5523D_Reg(0x30);
        bBuf &= (~0x01);
        Set_NCT5523D_Reg(0x30, bBuf);                           //Enable
WDTO
}
//---------------------------------------------------------------------------
```

```
void WDTEnable(unsigned char NewInterval)
{
        unsigned char bBuf;

        Set_NCT5523D_LD(0x08);                                  //switch
to logic device 8
        Set_NCT5523D_Reg(0x30, 0x01);                           //enable
timer

        bBuf = Get_NCT5523D_Reg(0xF0);
        bBuf &= (~0x08);
        Set_NCT5523D_Reg(0xF0, bBuf);                           //count
mode is second

        Set_NCT5523D_Reg(0xF1, NewInterval);        //set timer
}
//-----------------------------------------------------------------------
void WDTDisable(void)
{
        Set_NCT5523D_LD(0x08);                                  //switch
to logic device 8
        Set_NCT5523D_Reg(0xF1, 0x00);                           //clear
watchdog timer
        Set_NCT5523D_Reg(0x30, 0x00);
        //watchdog disabled
}
//-----------------------------------------------------------------------
```

3. Sample Code: The file NCT5523D.CPP

```
//---------------------------------------------------------------------------
//
// THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY
// KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
// IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR
// PURPOSE.
//
//---------------------------------------------------------------------------
#include "NCT5523D.H"
#include <dos.h>
//---------------------------------------------------------------------------
unsigned int NCT5523D_BASE;
void Unlock_NCT5523D (void);
void Lock_NCT5523D (void);
//---------------------------------------------------------------------------
unsigned int Init_NCT5523D(void)
{
        unsigned int result;
        unsigned char ucDid;

        NCT5523D_BASE = 0x4E;
        result = NCT5523D_BASE;

        ucDid = Get_NCT5523D_Reg(0x20);
        if (ucDid == 0xC4)
        //NCT5523D??
        {       goto Init_Finish;       }

        NCT5523D_BASE = 0x2E;
        result = NCT5523D_BASE;

        ucDid = Get_NCT5523D_Reg(0x20);
        if (ucDid == 0xC4)
        //NCT5523D??
        {       goto Init_Finish;       }

        NCT5523D_BASE = 0x00;
        result = NCT5523D_BASE;

Init_Finish:
        return (result);
}
//---------------------------------------------------------------------------
void Unlock_NCT5523D (void)
{
        outportb(NCT5523D_INDEX_PORT, NCT5523D_UNLOCK);
        outportb(NCT5523D_INDEX_PORT, NCT5523D_UNLOCK);
}
//---------------------------------------------------------------------------
```

```
void Lock_NCT5523D (void)
{
		outportb(NCT5523D_INDEX_PORT, NCT5523D_LOCK);
}
//-----------------------------------------------------------------------
void Set_NCT5523D_LD( unsigned char LD)
{
		Unlock_NCT5523D();
		outportb(NCT5523D_INDEX_PORT, NCT5523D_REG_LD);
		outportb(NCT5523D_DATA_PORT, LD);
		Lock_NCT5523D();
}
//-----------------------------------------------------------------------
void Set_NCT5523D_Reg( unsigned char REG, unsigned char DATA)
{
		Unlock_NCT5523D();
		outportb(NCT5523D_INDEX_PORT, REG);
		outportb(NCT5523D_DATA_PORT, DATA);
		Lock_NCT5523D();
}
//-----------------------------------------------------------------------
unsigned char Get_NCT5523D_Reg(unsigned char REG)
{
		unsigned char Result;
		Unlock_NCT5523D();
		outportb(NCT5523D_INDEX_PORT, REG);
		Result = inportb(NCT5523D_DATA_PORT);
		Lock_NCT5523D();
		return Result;
}
//-----------------------------------------------------------------------
```