IB899

Intel® Braswell

USER'S MANUAL

Version 1.2b

Acknowledgments

AMI BIOS is a trademark of American Megatrends Inc. PS/2 is a trademark of International Business Machines Corporation. Intel and Luna-Pier are registered trademarks of Intel Corporation. Microsoft Windows is a registered trademark of Microsoft Corporation. Winbond is a registered trademark of Winbond Electronics Corporation. All other product names or trademarks are properties of their respective owners.

Table of Contents

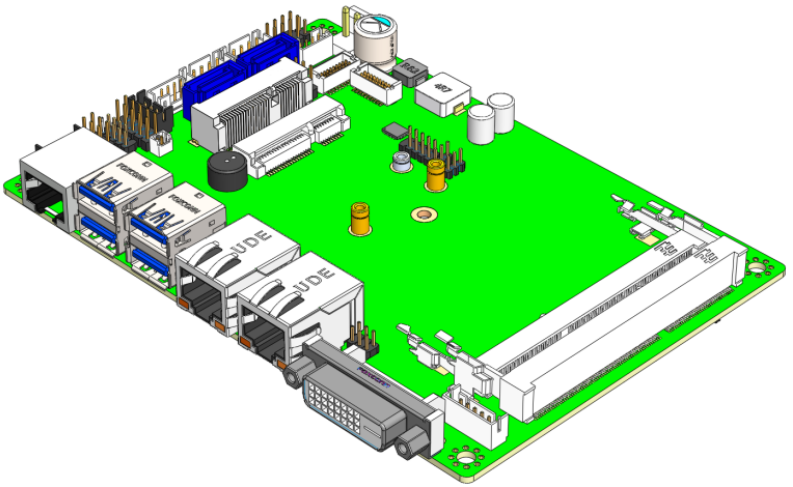
This page is intentionally left blank.

Introduction

IB899 is a 3.5-inch Disk Size SBC (102mm x 147mm) that comes with the Intel® Pentium® N3700 \ N3710 and Celeron® N3000 \ N3010 series processor. It has two DDR3L SO-DIMM sockets supporting up to 8GBs of system memory. It also features the Intel® Gen8-LP graphics engine with interface for CRT, DVI-D and 24-bit dual channel LVDS displays. The platform provides four connectors of USB 3.0 at the board edge and USB 2.0 two ports with pin header. Two SATA III ports are included.

IB899 Features

- 3.5-inch Disk Size SBC, 142mm x 102mm
- Onboard Intel® Pentium® N3700 \ N3710 series
  Celeron® N3000 \ N3010 series
- Two DDR3L SO-DIMM sockets, DDR3L-1600, Max. 8GB
- Intel® Gen8-LP graphics for CRT, DVI-D interface
- 24-bit dual channel LVDS interface
- Dual Intel I211-AT PCIe Gigabit LAN
- 4 x USB 3.0 on edge, 2 x USB 2.0 on board support
- Two SATA III, 2 x COM ports
- Digital I/O 4-in / 4-out, 1 x Full-size Mini-PCIe, 1 x Half-size Mini-PCIe
- Watchdog Timer, iSMART, RoHS compliance



Checklist

Your IB899 package should include the items listed below.

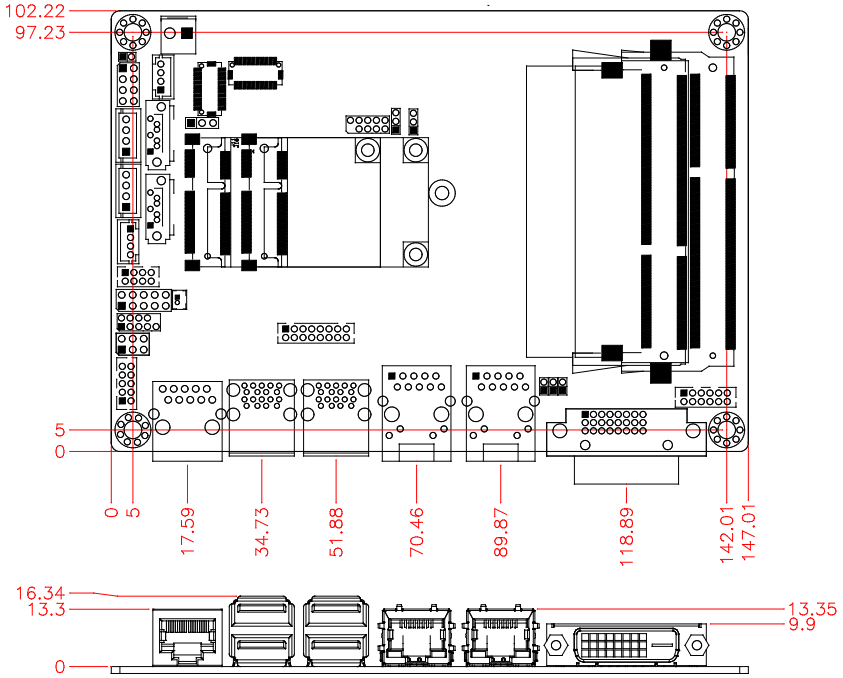
- 3.5-inch Disk Size SBC
- This User's Manual
- 1 CD containing chipset drivers and flash memory utility
- Serial ATA cable
- I/O shield

IB899 Specifications

Product Name	IB899F-370 (N3700 onboard) IB899F-371 (N3710 onboard) IB899F-300 (N3000 onboard) IB899F-301 (N3010 onboard) IB899A-370 (N3700 onboard, w/MicroSD & TPM 2.0) IB899A-371 (N3710 onboard, w/MicroSD & TPM 2.0) **Default silk screen model # on PCB is IB899F**
Form Factor	3.5"
SoC Type/Speed	14nm Technology Process, FCBGA1170, Package= 25mm x 27mm - Pentium® N3700 QC, up to 2.4Ghz , TDP=6W (C0-stepping, SR2A7) - **Pentium® N3710 QC, up to 2.56Ghz , TDP=6W (D1-stepping, SR2KL)** - Celeron® N3000 DC, up to 2.08GHz, TDP=4W (C0-stepping, SR29J) - **Celeron® N3010 DC, up to 2.24GHz, TDP=4W (D1-stepping, SR2KM)**
Cache	2MB
BIOS	AMI BIOS
Memory	Intel® Pentium®/Celeron® SoC integrated memory controller 2 x DDR3 SO-DIMM socket, Support DDR3L (1.35V) only Max. memory to 8GB (Non-ECC, Un-buffered)
VGA	Intel® Pentium®/Celeron® SoC integrated Gen 8 graphics Support 3 x independent displays, DX11.1 (DX12 for Win 10) & OpenGL 4.2 - VGA x 1: Thru NXP PTN3392 DP to VGA **1920x1200 @ 60 Hz** (C0133392BS**F4**15000P) - DVI-D x 1 ** 2560x1600@ 60 Hz** - LVDS: 24-bit dual channel via NXP PTN3460 thru eDP **1920x1200 @ 60 Hz** (C01Z3460BS**F6**12000P)
LAN	Intel® I211-AT PCIe Gigabit LAN x 2
USB (Universal Serial Bus)	Intel® Pentium®/Celeron® SoC built-in USB 3.0 host controller, support **4** ports - Edge I/O x 4 Intel® Pentium®/Celeron® SoC built-in USB 2.0 host controller with USB hub Renesas uPD720115K8-611-BAK-A(C013720115K861000P) For total 4 ports x USB 2.0 - 2 Ports thru pin header - 2 Ports thru Mini PCIe slot
Serial ATA Ports	Intel® Pentium®/Celeron® SoC built-in SATA III controller, supports **2** ports
Audio	Intel® Pentium®/Celeron® SoC built-in HD Audio controller + Realtek ALC662 Codec
LPC I/O	Nuvoton NCT5523D, Ver. C [64-pin LQFP, 7x7x1.4mm] COM #1 (RS232/422/485) - With EXAR SP339 (C014339EER1L29000P) x 1 for jumper-less - Support ring-in with power @500 mA (selectable for 5V or 12V) COM #2 (RS-232 only) [Hardware Monitor] 2 x Thermal inputs; 2 x Voltage monitoring
Digital IO	4 in & 4 out
Expansion Slots	Mini PCIe socket x **2** Full-sized x1 : with USB 2.0 signal + support mSATA(via NXP switch) Half-sized x1: with USB 2.0 signal
Edge Connectors	DVI-D x 1 for DVI COM x 1 (RJ45 connector type) RJ45 x 2 for LAN 1 & 2 USB 3.0 stack connector x 1 for 2 ports (from SoC) USB 3.0 stack connector x 1 for 2 ports (from SoC)

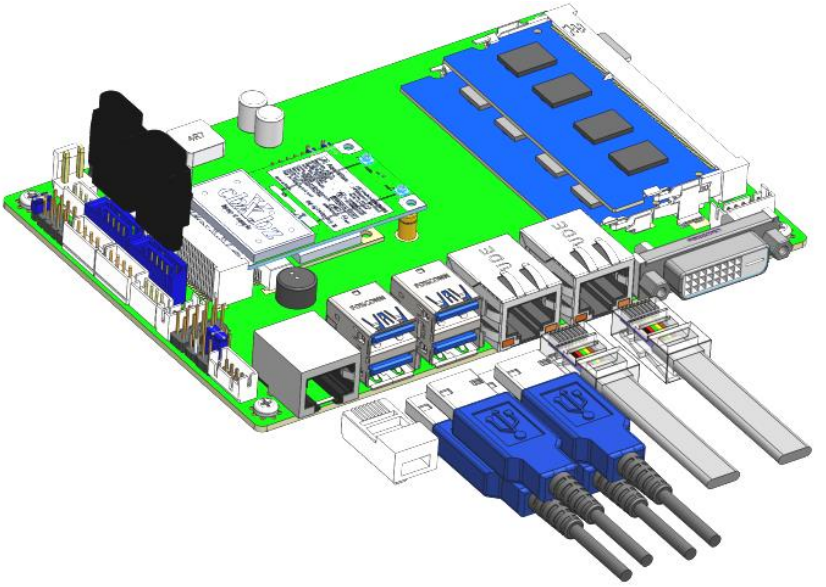
Onboard Header/ Connector	DF11 2 x 4 pins header x 1 for 2 x USB 2.0 DF20 socket connector x 2 for 24-bit dual channel LVDS (C1220220020220300P) 4 pins box header x 1 for backlight/brightness control (PWM mode) DF11 2 x 6 pins box header x1 for Audio DF11 2 x 8 pins box header x 1 for VGA DF11 2 x 5 pins box header x 1 for COM2 2 x 5 pins header x 1 for LPC(80-port card debugging purpose) Mini PCI-e(1x) connector x 2 SATA III connector x 2 for SATA device(BLUE color) 4-pins power connector x 2 (JST type, For SATA III device) 2 x 4 pins pin-header x 1 for front I/O (RST, PWR, LEDs) 2-pins connector x 1 for power input (C12101112202101HAP) Micro SD slot x 1 @ solder side (IB899A only) BR2032 w/ adhesive tape = C272113012032B100P+ M1401020010010100
Watchdog Timer	Yes (256 segments, 0, 1, 2…255 sec/min)
DC Input	9V~+24V (±5 % tolerance)
TPM 2.0	Infineon SL9665 (IB899A-370 only) [C01Z9665TT2007000P] Infineon SL9665 (IB899A-371 only) [C01Z9665TT2007000P]
Others	iSMART 3.2
OS support	**** Please note the priority**** - Windows 8.1(64-bit) - Windows 7(64-bit,refer to Intel #558302 EHCI OS USB installation guide) - Fedora (Installation) - Ubuntu (Installation) - Windows 10 (32-bit / 64-bit) - Embedded OS will be test by request
RoHS / REACH/ LVD /CE	Yes / Yes / Yes / Class B
Operating system	0° C to +60° C
Board Size	102mm x 147mm

Board Dimensions



Installations

This section provides information on how to use the jumpers and connectors on the IB899 in order to set up a workable system. The topics covered are:



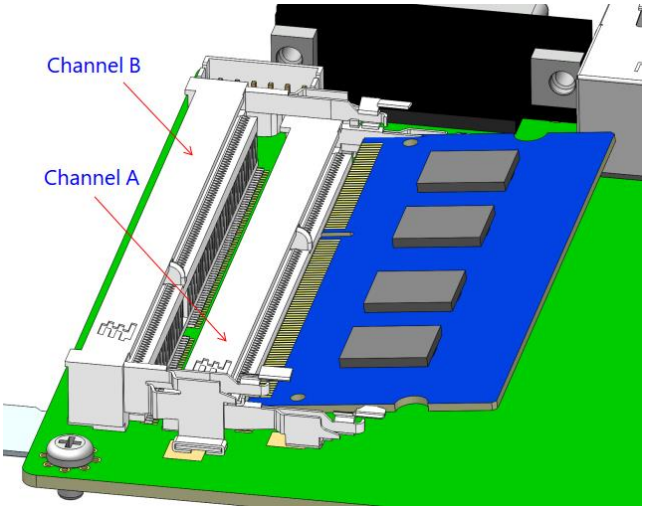
Installing the Memory

The IB899 board supports two DDR3L-1600 memories.

Installing and Removing Memory Modules

To install the DDR3 modules, locate the memory slot on the board and perform the following steps:

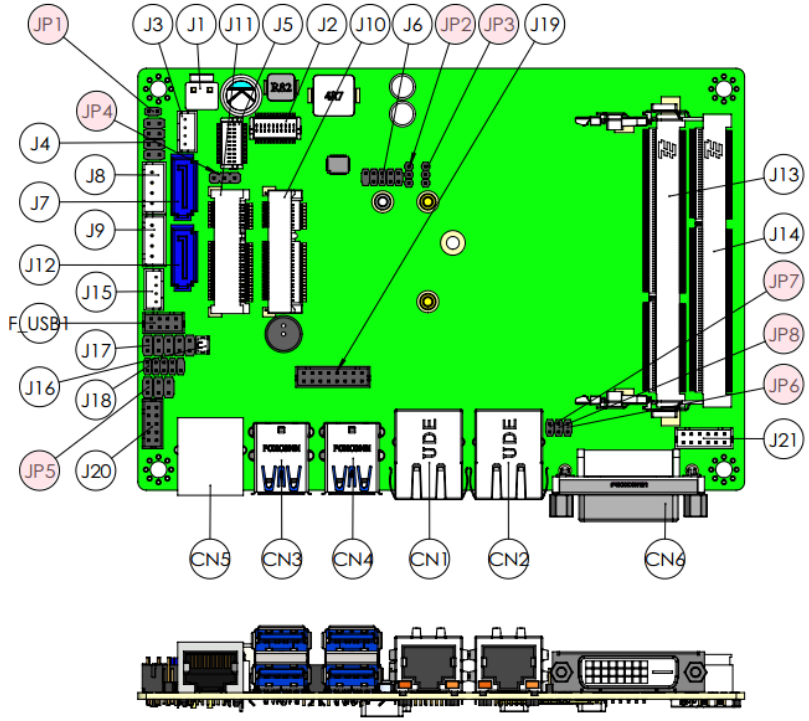
1. Hold the DDR3 module so that the key of the DDR3 module aligned with that on the memory slot.
2. Gently push the DDR3 module in an upright position until the clips of the slot close to hold the DDR3 module in place when the DDR3 module touches the bottom of the slot.
3. To remove the DDR3 module, press the clips with both hands.
4. You must install the DDR3 module on Channel A when you have only one SD-DIMM memory module.



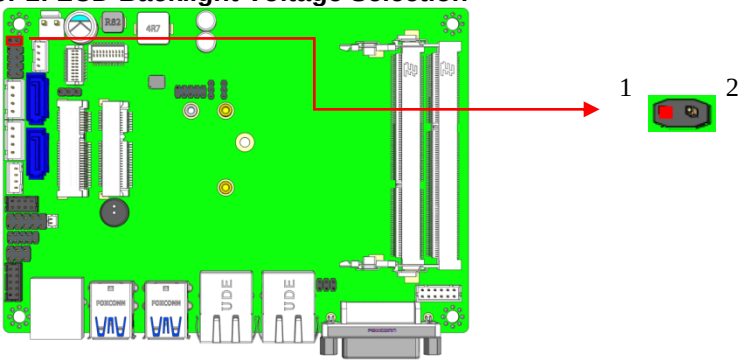
Setting the Jumpers

Jumpers are used on IB899 to select various settings and features according to your needs and applications. Contact your supplier if you have doubts about the best configuration for your needs. The following lists the connectors on IB899 and their respective functions.

Jumper Locations on IB899

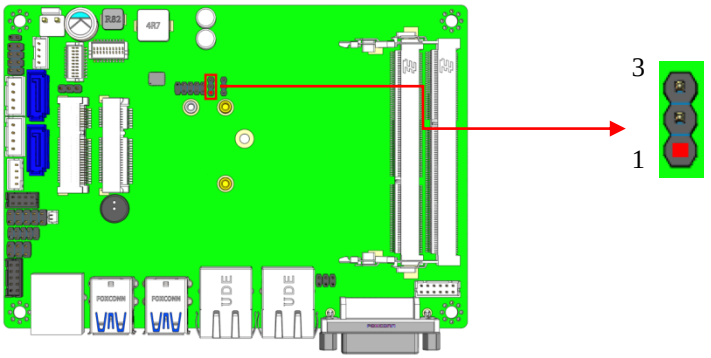


JP1: LCD Backlight Voltage Selection



JP1	Setting
	+3.3V
	+5V

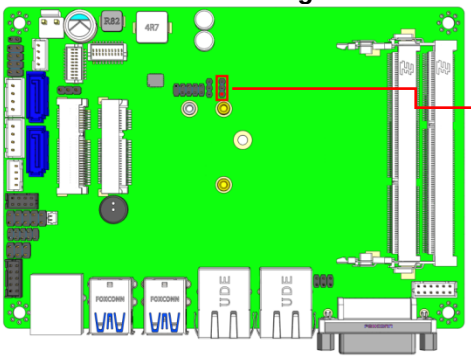
JP2: Clear RTC

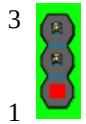


JP2	Setting
	Save RTC*
	Clear RTC

*: Default Setting

JP3: Clear CMOS Setting

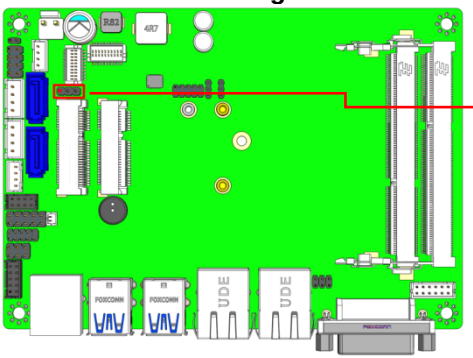




JP3	Setting
	Save CMOS*
	Clear CMOS

*: Default Setting

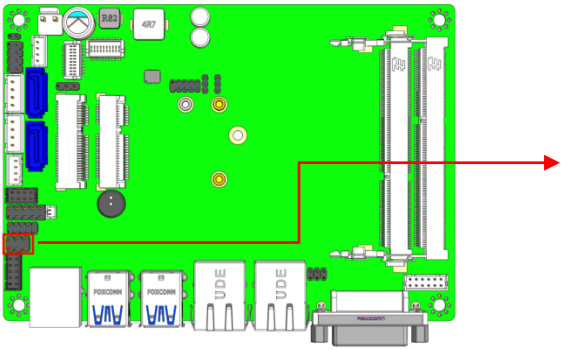
JP4: LCD Panel Voltage Selection



JP4	LCD Panel Power
	+3.3V*
	+5V

*: Default Setting

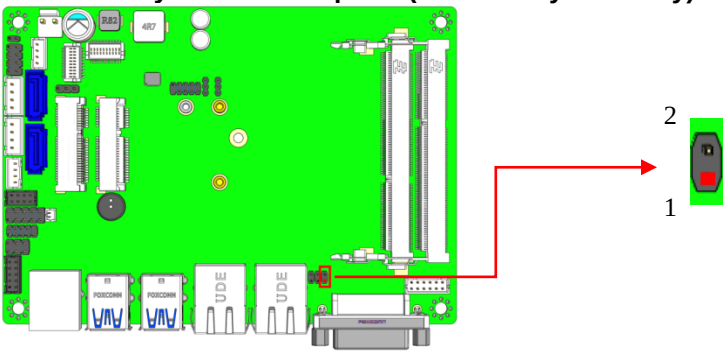
JP5: COM1 Ring Pin Function Selection



JP5	Setting
	+12V
	Ring*
	+5V

*: Default Setting

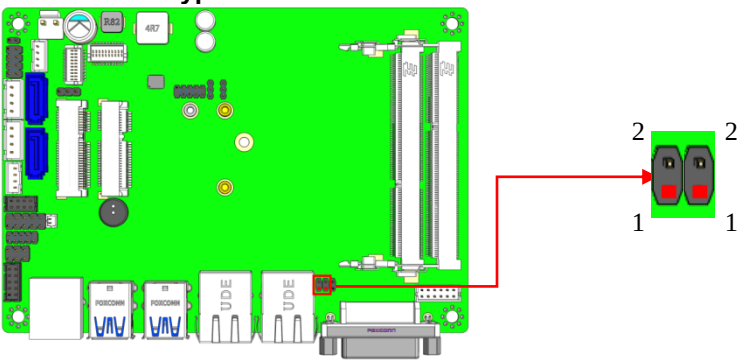
JP6: Security Flash Descriptors (For Factory Use Only)



JP6	Setting
	Open*
	Close

*: Default Setting

JP7/JP8: OS Type Selection



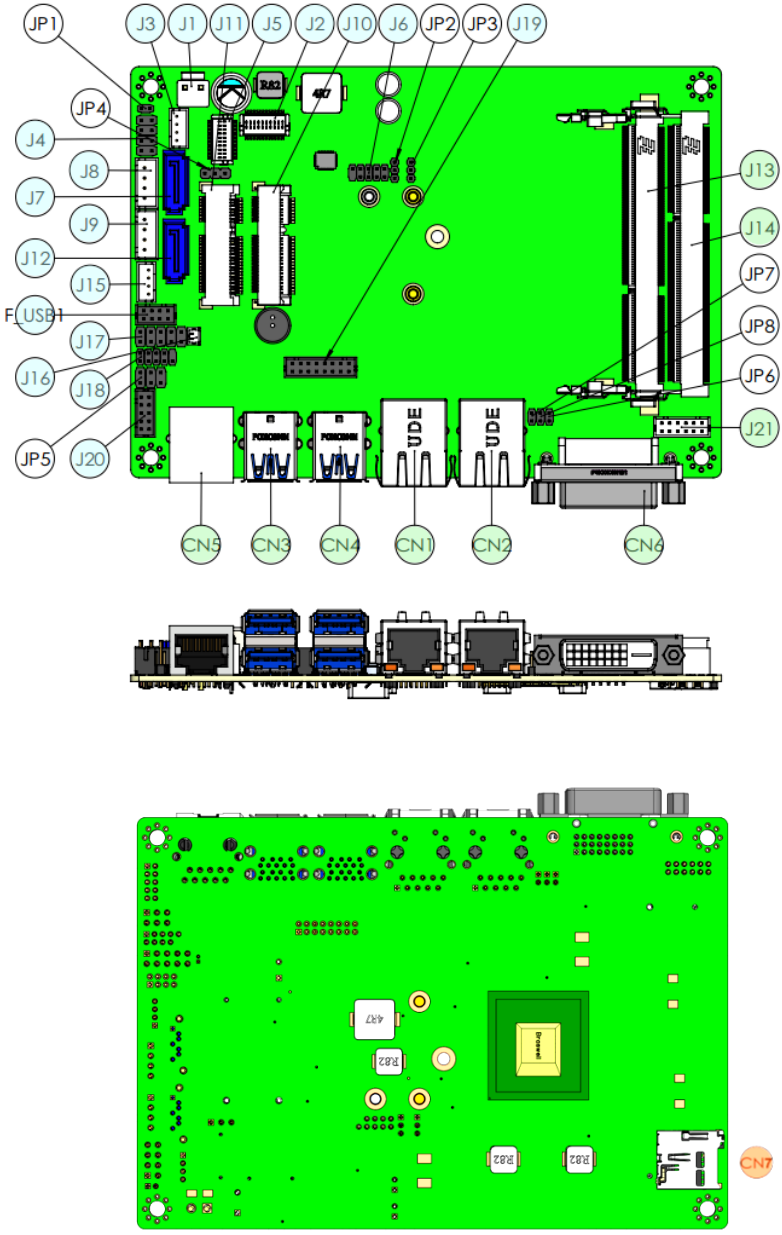
JP7/JP8	Setting
	Win8/Android
	Linux
	DOS, Win 7 Legacy
	N/A

*: Default Setting

Connectors on IB899

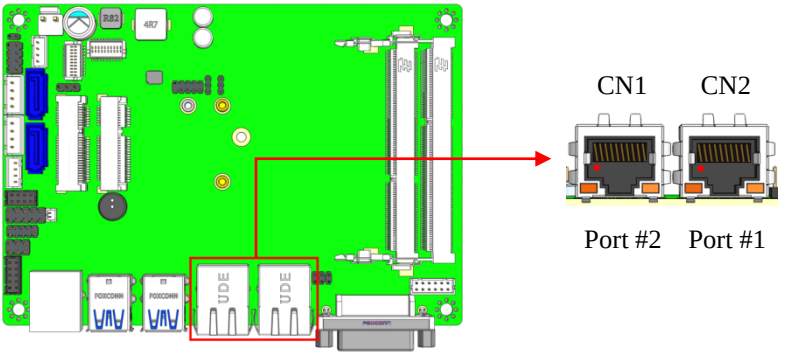
The connector on IB899 allows you to connect external devices such as keyboard, floppy disk drives, hard disk drives etc. The following table lists the connectors on IB899 and their respective functions.

Connector Locations on IB899



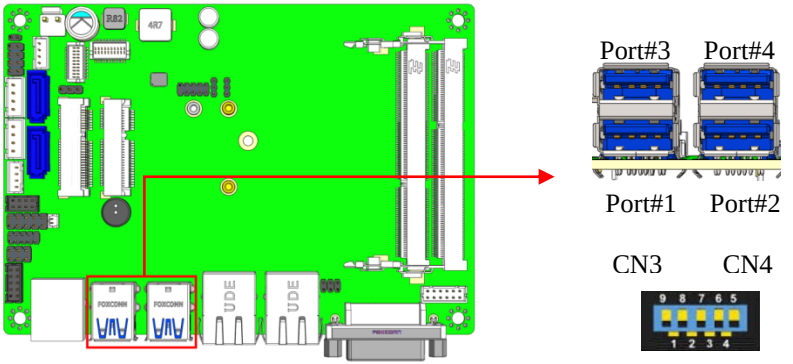
CN1/CN2: Gigabit LAN Port #2 / #1 Connectors

CN1: Port #2 **CN2: Port #1**



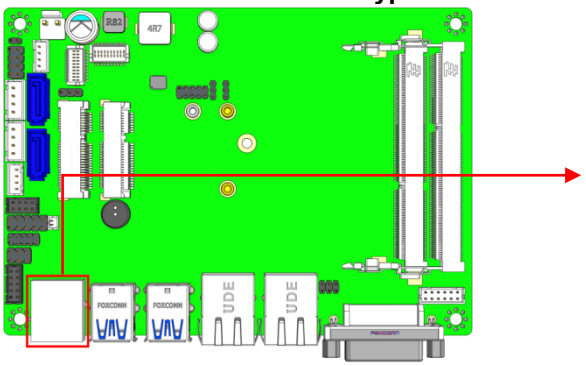
Signal Name	Pin #	Pin #	Signal Name
TX+D1	1	2	TX-D1
RX+D2	3	4	BI+D3
BI-D3	5	6	RX-D2
BI+D4	7	8	BI-D4

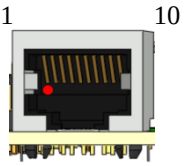
CN3/CN4: USB 3.0 Connectors



Signal Name	Pin #	Pin #	Signal Name
VBUS	1	2	DATA-
DATA+	3	4	GND
SSRX-	5	6	SSRX+
GND	7	8	SSTX-
SSTX+	9		

CN5: COM1 in RJ45 10P10C Type Connector





PIN Assignment for RS232 mode

Signal Name	Pin #	Pin #	Signal Name
DSR#	1	2	GND
GND	3	4	TX
RX	5	6	DCD#
DTR#	7	8	CTS#
RTS#	9	10	RI#

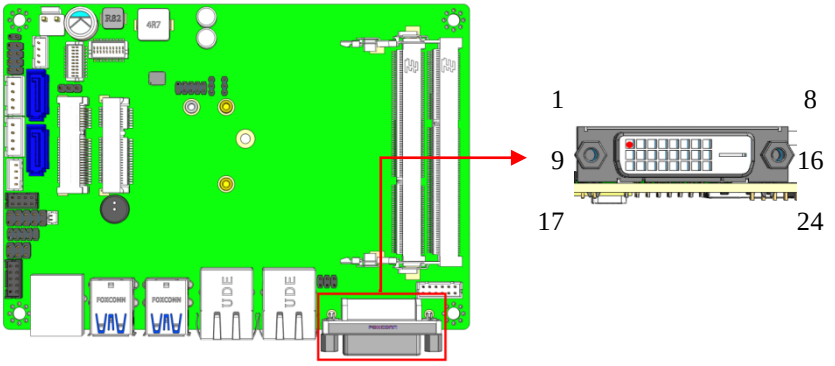
PIN Assignment for RS422 mode

Signal Name	Pin #	Pin #	Signal Name
-	1	2	-
-	3	4	RX+
TX+	5	6	TX-
RX-	7	8	-
-	9	10	-

PIN Assignment for RS485 mode

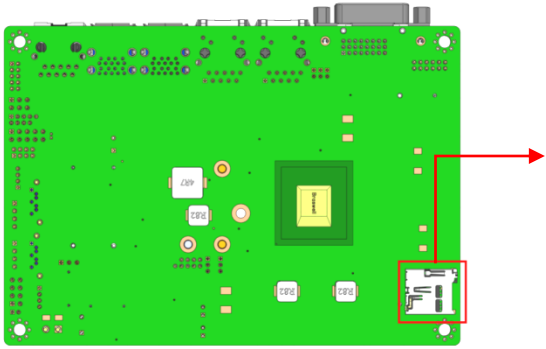
Signal Name	Pin #	Pin #	Signal Name
-	1	2	-
-	3	4	-
DATA+	5	6	DATA-
-	7	8	-
-	9	10	-

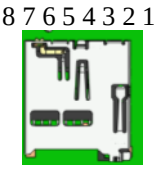
CN6: DVI-D Connector



Signal Name	Pin #	Pin #	Signal Name
TMDS DAT2-	1	2	TMDS DAT2+
TMDS 2/4 SHIELD	3	4	TMDS DAT4-
TMDS DAT4+	5	6	DDC CLOCK
DDC DATA	7	8	ANALOG VSYNC
TMDS DAT1-	9	10	TMDS DAT1+
TMDS 1/3 SHIELD	11	12	TMDS DAT3-
TMDS DAT3+	13	14	+5V POWER
GND (for +5V)	15	16	HOT PLUG DET
TMDS DAT0-	17	18	TMDS DAT0+
TMDS 0/5 SHIELD	19	20	TMDS DAT5-
TMDS DAT5+	21	22	TMDS CLK SHIELD
TMDS CLOCK+	23	24	TMDS CLOCK-

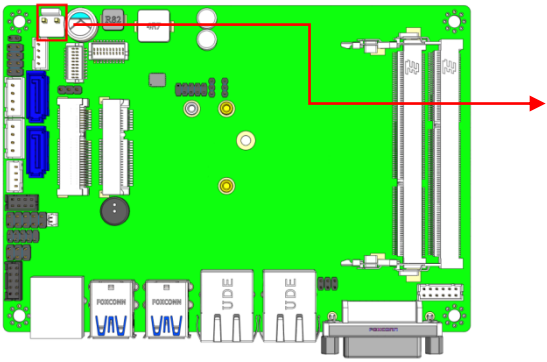
CN7: Micro SD Connector (On Bottom Side)





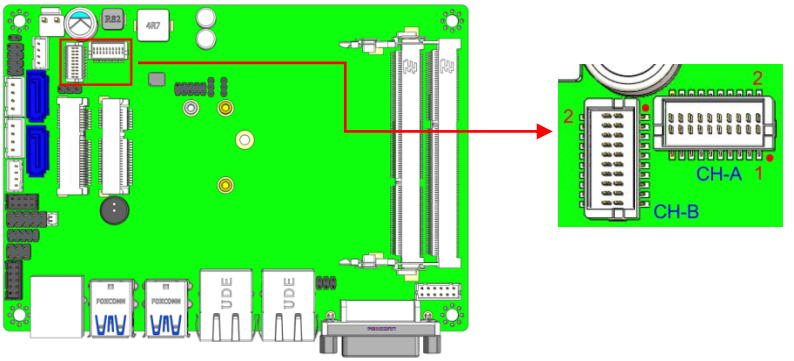
Pin #	Signal Name
1	DAT2
2	CD/DAT3
3	CMD
4	VDD
5	CLK
6	VSS
7	DAT0
8	DAT1

J1: Power Input Connector



Pin #	Signal Name
1	POWER+
2	GND

J2/J5: Dual Channel LVDS Connector



J2 (Channel A) Pin Assignment

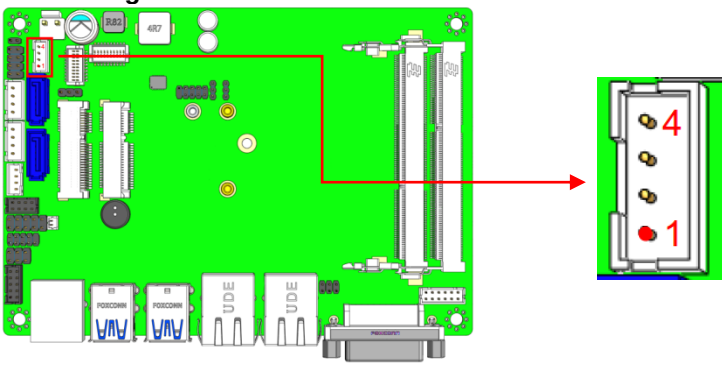
Signal Name	Pin #	Pin #	Signal Name
LVDS_SAO_P	1	2	LVDS_SAO_N
GND	3	4	GND
LVDS_SBO_P	5	6	LVDS_SBO_N
GND	7	8	GND
LVDS_SCO_P	9	10	LVDS_SCO_N
GND	11	12	GND
LVDS_SCKO_P	13	14	LVDS_SCKO_N
GND	15	16	GND
LVDS_SDO_P	17	18	LVDS_SDO_N
VDD	19	20	VDD

J5 (Channel B) Pin Assignment

Signal Name	Pin #	Pin #	Signal Name
LVDS_SAE_P	1	2	LVDS_SAE_N
GND	3	4	GND
LVDS_SBE_P	5	6	LVDS_SBE_N
GND	7	8	GND
LVDS_SCE_P	9	10	LVDS_SCE_N
GND	11	12	GND
LVDS_SCKE_P	13	14	LVDS_SCKE_N
GND	15	16	GND
LVDS_SDE_P	17	18	LVDS_SDE_N
VDD	19	20	VDD

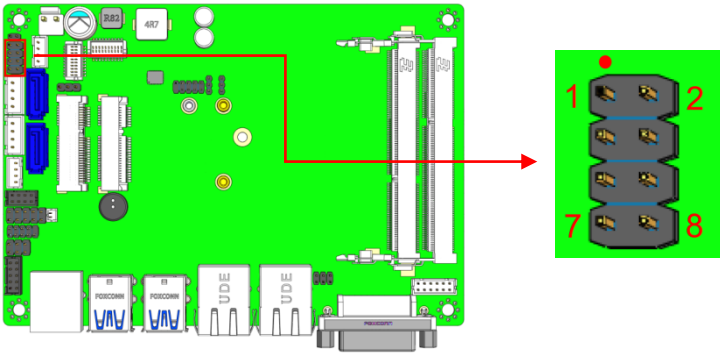
*** J2 /J5 connector model: HIROSE DF20G-20DP-1V(56) ***
https://www.hirose.com/product/en/products/DF20/DF20F-20DP-1V(56)/

J3: Backlight LCD Panel Power Connector



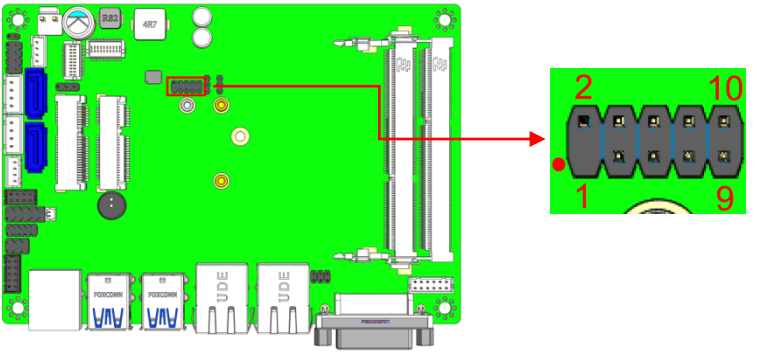
Pin #	Signal Name
1	VCC12
2	LVDS_BLON#
3	BKL_CTRL
4	GND

J4: Front Panel Pin Header



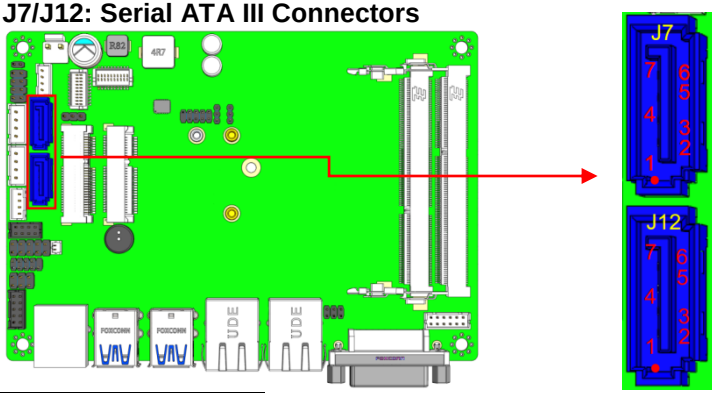
Signal Name	Pin #	Pin #	Signal Name
POWER BUTTON -	1	2	POWER BUTTON +
HDD LED +	3	4	HDD LED -
RESET BUTTON -	5	6	RESET BUTTON +
POWER LED +	7	8	POWER LED -

J6: SPI Flash Burn-in Pin Header



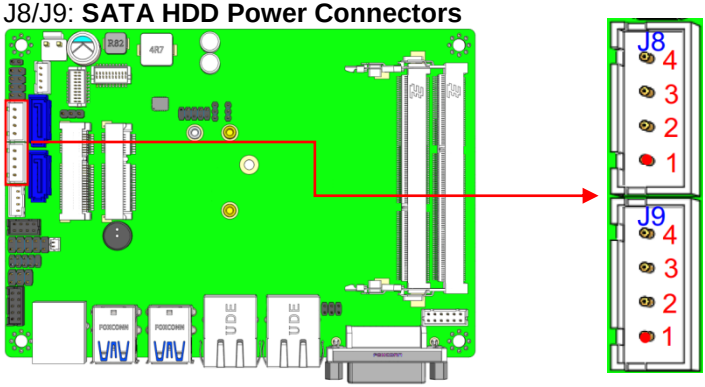
Signal Name	Pin #	Pin #	Signal Name
- (No Pin)		2	NC
SPI CS	3	4	+1.8V
SPI SO	5	6	SPI HOLD#
SPI WP#	7	8	SPI CLK
GND	9	10	SPI SI

J7/J12: Serial ATA III Connectors



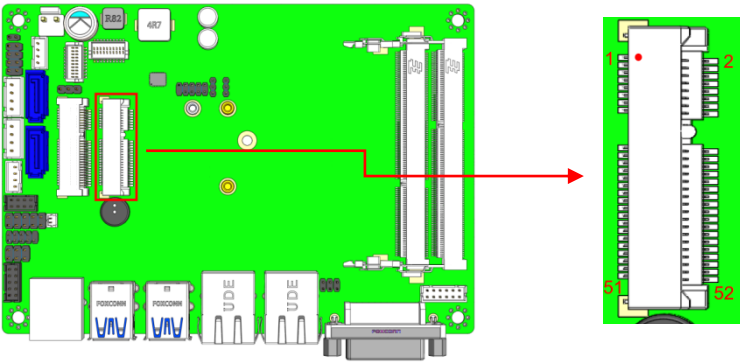
Pin #	Signal Name
1	GND
2	TXP0
3	TXN0
4	GND
5	RXN0
6	RXP0
7	GND

J8/J9: **SATA HDD Power Connectors**

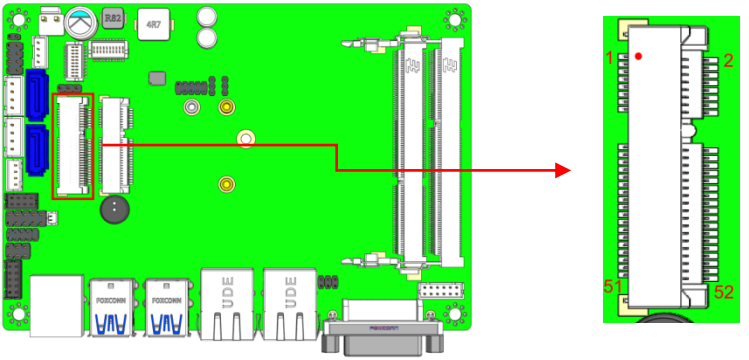


Pin #	Signal Name
1	+5V
2	GND
3	GND
4	+12V

J10: Half-sized Mini PCIe Sockets



J11: Full-sized Mini PCIe and mSATA Socket



Notes:
For manufacturers or system integrators, the exciting benefit is they can have a circuit board with co-layout design of Mini PCIe and mSATA. This feature allows users to enjoy wider applications by installing optional Mini PCIe or mSATA in the same placement site.
I

Half-sized Mini PCIe Socket Pin Assignments:

Signal Name	Pin #	Pin #	Signal Name
PCIe_WAKE#	1	2	3.3V_AUX
COEX1	3	4	GND
COEX2	5	6	+1.5V
CLKREQ#	7	8	UIM_PWR
GND	9	10	UIM_DATA
PCIe_REFCLK2-	11	12	UIM_CLK1
PCIe_REFCLK2+	13	14	UIM_RESET
GND	15	16	UIM_CLK2
RSVD (UIM)	17	18	GND
RSVD (UIM)	19	20	W_DISABLE
PCIe_PER_2-	21	22	PERST#
PCIe_PER_2+	23	24	3.3V_AUX
GND	25	26	GND
GND	27	28	+1.5V
PCIe_PET_2-	29	30	SMB_CLK
PCIe_PET_2+	31	32	SMB_DAT
GND	33	34	GND
GND	35	36	USB_D3-
3.3V_AUX	37	38	USB_D3+
3.3V_AUX	39	40	GND
GND	41	42	LED_WWAN#
RSVD	43	44	LED_WLAN#
RSVD	45	46	LED_WPAN#
RSVD	47	48	+1.5V
RSVD	49	50	GND
RSVD	51	52	3.3V_AUX

Notes:

- Pin 3,5,17,19,43,45,47,49,51,8,10,12,14,16,42,44,46 all these pins are no connected pin. (NC)
- Pin 7, 20 just only pulled up to 3.3V through a 10Kohm resistor.
- 3.3V_AUX is a 3.3V standby power, it means always has power on this pin when the adaptor is plugged into the mainboard.

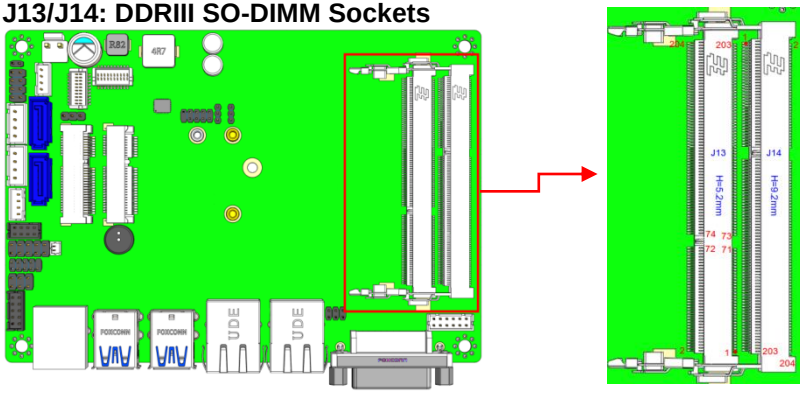
Full-sized Mini PCIe and mSATA Socket Pin Assignments:

Signal Name	Pin #	Pin #	Signal Name
PCIe_WAKE#	1	2	3.3V_AUX
COEX1	3	4	GND
COEX2	5	6	+1.5V
CLKREQ#	7	8	UIM_PWR
GND	9	10	UIM_DATA
PCIe_REFCLK3-	11	12	UIM_CLK1
PCIe_REFCLK3+	13	14	UIM_RESET
GND	15	16	UIM_CLK2
RSVD (UIM)	17	18	GND
RSVD (UIM)	19	20	W_DISABLE
PCIe_PER_3- / SATA_RXP1	21	22	PERST#
PCIe_PER_3+ / SATA_RXN1	23	24	3.3V_AUX
GND	25	26	GND
GND	27	28	+1.5V
PCIe_PET_3- / SATA_TXN1	29	30	SMB_CLK
PCIe_PET_3+ / SATA_TXP1	31	32	SMB_DAT
GND	33	34	GND
GND	35	36	USB_D4-
3.3V_AUX	37	38	USB_D4+
3.3V_AUX	39	40	GND
GND	41	42	LED_WWAN#
RSVD	43	44	LED_WLAN#
RSVD	45	46	LED_WPAN#
RSVD	47	48	+1.5V
RSVD	49	50	GND
mSATA#/mPCIe Selection Pin	51	52	3.3V_AUX

Notes:

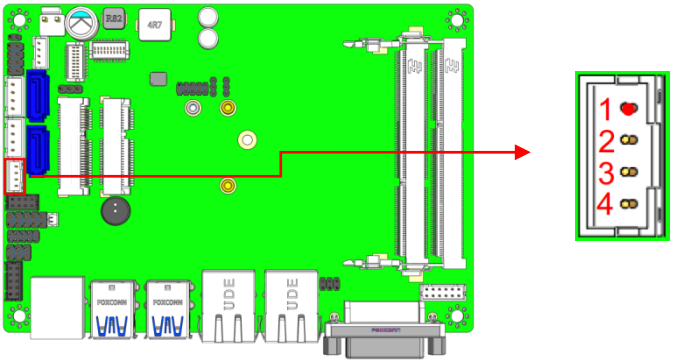
- Pin 3,5,17,19,43,45,47,49,51,8,10,12,14,16,42,44,46 all these pins are no connected pin. (NC)
- Pin 7, 20 just only pulled up to 3.3V through a 10Kohm resistor.
- 3.3V_AUX is a 3.3V standby power, it means always has power on this pin when the adaptor is plugged into the mainboard.
- Pin 51 is used for automatically select card type.
  mSATA card this pin is low, Mini PCIe card this pin is high.

J13/J14: DDRIII SO-DIMM Sockets



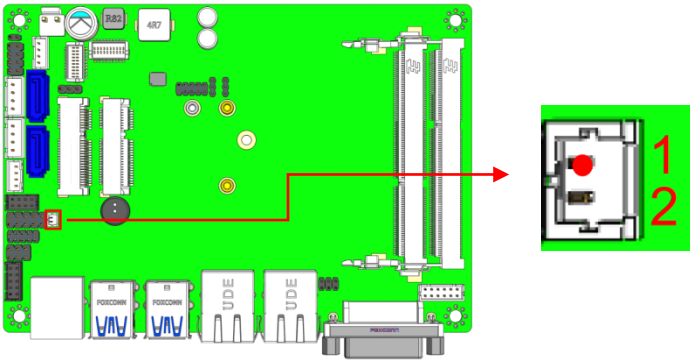
1	V_{REFDQ}	2	V_{SS}	53	DQ19	54	V_{SS}	105	V_{DD}	106	V_{DD}	155	V_{SS}	156	V_{SS}
3	V_{SS}	4	DQ4	55	V_{SS}	56	DQ28	107	A10/AP	108	BA1	157	DQ42	158	DQ46
5	DQ0	6	DQ5	57	DQ24	58	DQ29	109	BA0	110	$\overline{RAS}$	159	DQ43	160	DQ47
7	DQ1	8	V_{SS}	59	DQ25	60	V_{SS}	111	V_{DD}	112	V_{DD}	161	V_{SS}	162	V_{SS}
9	V_{SS}	10	$\overline{DQS0}$	61	V_{SS}	62	$\overline{DQS3}$	113	$\overline{WE}$	114	$\overline{S0}$	163	DQ48	164	DQ52
11	DM0	12	DQS0	63	DM3	64	DQS3	115	$\overline{CAS}$	116	ODT0	165	DQ49	166	DQ53
13	V_{SS}	14	V_{SS}	65	V_{SS}	66	V_{SS}	117	V_{DD}	118	V_{DD}	167	V_{SS}	168	V_{SS}
15	DQ2	16	DQ6	67	DQ26	68	DQ30	119	A13/NC	120	ODT1	169	$\overline{DQS6}$	170	DM6
17	DQ3	18	DQ7	69	DQ27	70	DQ31	121	$\overline{ST}$	122	NC	171	DQS6	172	V_{SS}
19	V_{SS}	20	V_{SS}	71	V_{SS}	72	V_{SS}	123	V_{DD}	124	V_{DD}	173	V_{SS}	174	DQ54
21	DQ8	22	DQ12	73	CKE0	74	CKE1	125	NC	126	V_{REFCA}	175	DQ50	176	DQ55
23	DQ9	24	DQ13	75	V_{DD}	76	V_{DD}	127	V_{SS}	128	V_{SS}	177	DQ51	178	V_{SS}
25	V_{SS}	26	V_{SS}	77	NC	78	NC	129	DQ32	130	DQ36	179	V_{SS}	180	DQ60
27	$\overline{DQS1}$	28	DM1	79	BA2	80	NC	131	DQ33	132	DQ37	181	DQ56	182	DQ61
29	DQS1	30	$\overline{RESET}$	81	V_{DD}	82	V_{DD}	133	V_{SS}	134	V_{SS}	183	DQ57	184	V_{SS}
31	V_{SS}	32	V_{SS}	83	A12/$\overline{BC}$	84	A11	135	$\overline{DQS4}$	136	DM4	185	V_{SS}	186	$\overline{DQS7}$
33	DQ10	34	DQ14	85	A9	86	A7	137	DQS4	138	V_{SS}	187	DM7	188	DQS7
35	DQ11	36	DQ15	87	V_{DD}	88	V_{DD}	139	V_{SS}	140	DQ38	189	V_{SS}	190	V_{SS}
37	V_{SS}	38	V_{SS}	89	A8	90	A6	141	DQ34	142	DQ39	191	DQ58	192	DQ62
39	DQ16	40	DQ20	91	A5	92	A4	143	DQ35	144	V_{SS}	193	DQ59	194	DQ63
41	DQ17	42	DQ21	93	V_{DD}	94	V_{DD}	145	V_{SS}	146	DQ44	195	V_{SS}	196	V_{SS}
43	V_{SS}	44	V_{SS}	95	A3	96	A2	147	DQ40	148	DQ45	197	SA0	198	$\overline{EVENT}$
45	$\overline{DQS2}$	46	DM2	97	A1	98	A0	149	DQ41	150	V_{SS}	199	V_{DDSPD}	200	SDA
47	DQS2	48	V_{SS}	99	V_{DD}	100	V_{DD}	151	V_{SS}	152	$\overline{DQS5}$	201	SA1	202	SCL
49	V_{SS}	50	DQ22	101	CK0	102	CK1	153	DM5	154	DQS5	203	Vtt	204	Vtt
51	DQ18	52	DQ23	103	$\overline{CK0}$	104	$\overline{CK1}$								

J15: MCU Burn-in Connector



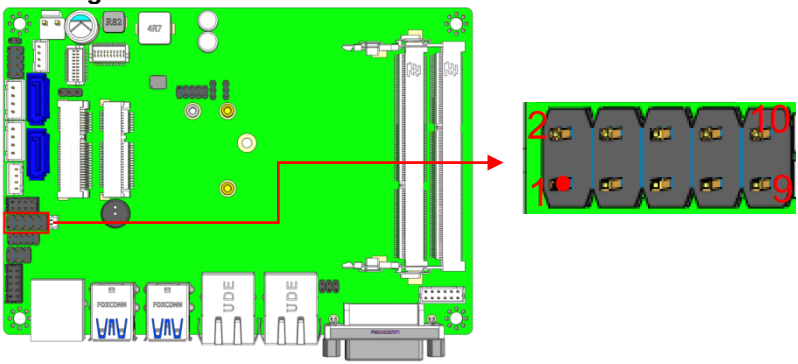
Pin #	Signal Name
1	3.3V
2	SBW_TCK
3	SBW_TDIO
4	GND

J16: Battery Connector



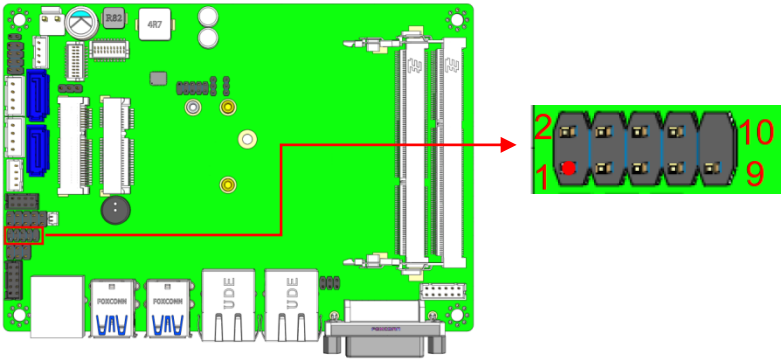
Pin #	Signal Name
1	BT+
2	GND

J17: Digital I/O Connector



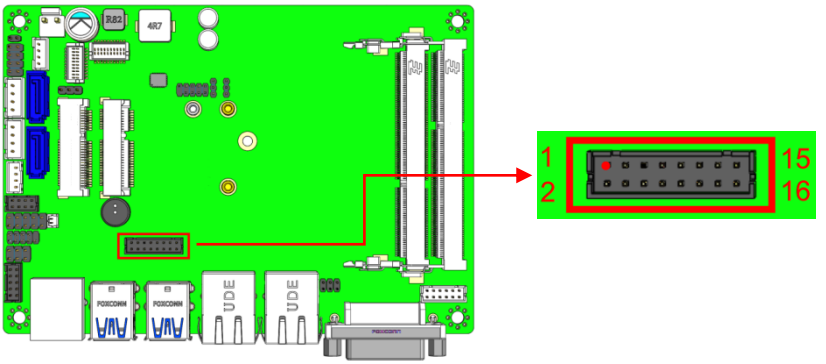
Signal Name	Pin #	Pin #	Signal Name
GND	1	2	+5V
OUT3	3	4	OUT1
OUT2	5	6	OUT0
IN3	7	8	IN1
IN2	9	10	IN0

J18: LPC Debug Connector



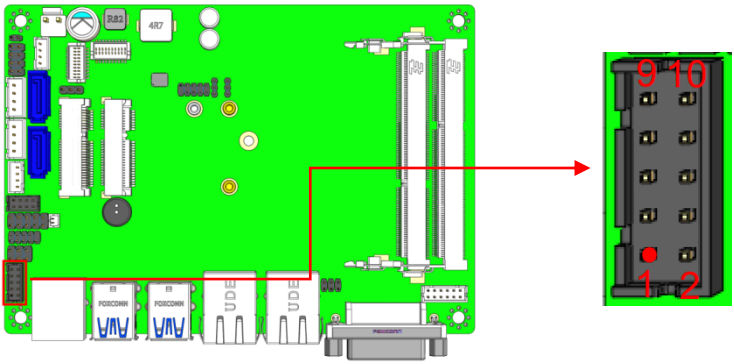
Signal Name	Pin #	Pin #	Signal Name
LPC AD0	1	2	RESET#
LPC AD1	3	4	LPC FRAME#
LPC AD2	5	6	+3.3V
LPC AD3	7	8	GND
LPC CLK	9	10	Empty Pin

J19: VGA Connector



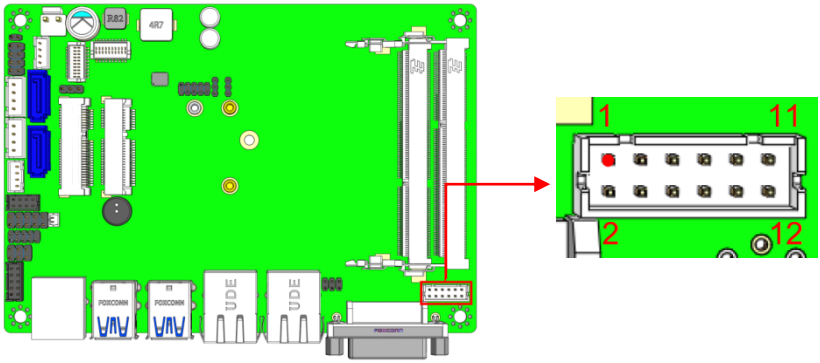
Signal Name	Pin #	Pin #	Signal Name
CRT RED	1	2	+5V
CRT GREEN	3	4	GND
CRT BLUE	5	6	NC
NC	7	8	DDC DATA
GND	9	10	CRT HSYNC
GND	11	12	CRT VSYNC
GND	13	14	DDC CLK
GND	15	16	NC

J20: COM2 Connector



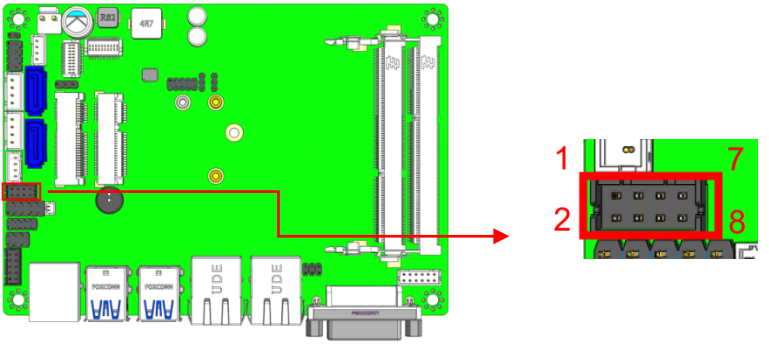
Signal Name	Pin #	Pin #	Signal Name
COM2 DCD#	1	2	COM2 RX
COM2 TX	3	4	COM2 DTR#
COM2 GND	5	6	COM2 DSR#
COM2 RTS#	7	8	COM2 CTS#
COM2 RI#	9	10	

J21: Audio Connector



Signal Name	Pin #	Pin #	Signal Name
LINE-OUT LEFT	1	2	LINE-OUT RIGHT
LINE-OUT JACK DET	3	4	GND
LINE-IN LEFT	5	6	LINE-IN RIGHT
LINE-IN JACK DET	7	8	GND
MIC LEFT	9	10	MIC RIGHT
MIC JACK DET	11	12	GND

F_USB1: USB2.0 Connector



Signal Name	Pin #	Pin #	Signal Name
GND	2	1	+5V
USB2_P2D+	4	3	USB2_P1D-
USB2_P2D-	6	5	USB2_P1D+
+5V	8	7	GND

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:

BIOS Introduction

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

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 are immediately activated. Pressing 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 wish 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:

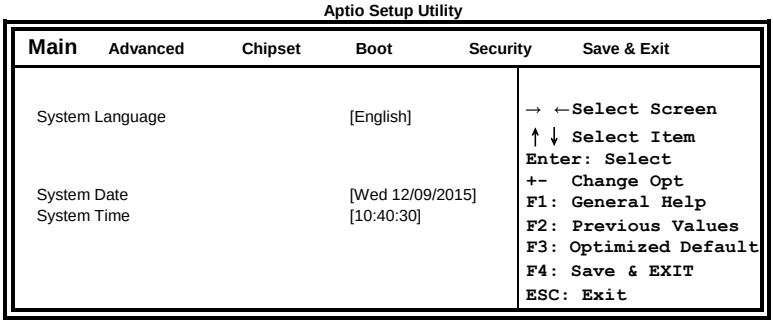
```
Press  <DEL>  to  Enter  Setup
```

In general, you 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 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 cause the system to become unstable and crash in some cases.*

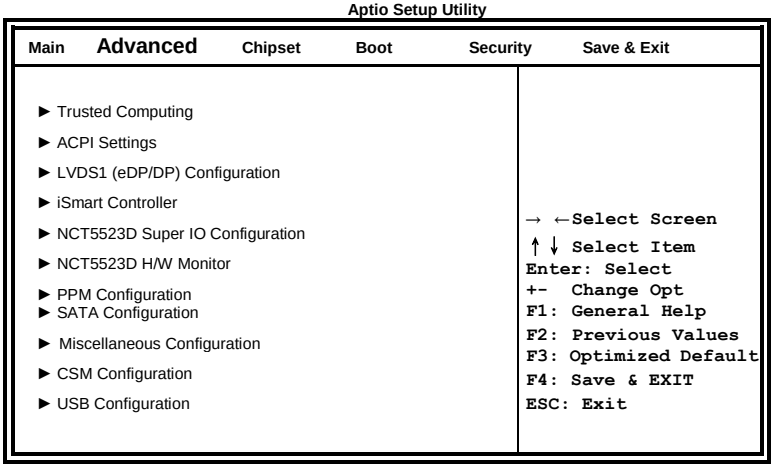
Main Settings



System Date
Set the Date. Use Tab to switch between Data elements.

System Time
Set the Time. Use Tab to switch between Data elements.

Advanced Settings



Trusted Computing

Aptio Setup Utility

Main	Advanced	Chipset	Boot	Security	Save & Exit
TPM20 Device Found					
Security Device Support			Enable		
Pending operation			None		
Platform Hierarchy			Enabled		
Storage Hierarchy			Enabled		
Endorsement Hierarchy			Enabled		
HashPolicy			Sha-1		
TPM 20 InterfaceType			CRB		
Device Select			Auto		

→ ←Select Screen
↑ ↓ Select Item
Enter: Select
+- Change Opt.
F1: General Help
F2: Previous Values
F3: Optimized Defaults
F4: Save & Exit
ESC: Exit

Security Device Support

Enables or Disables BIOS support for security device. O.S. will not show Security Device. TCG EFI protocol and INT1A interface will not be available.

Pending operation

Schedule an Operation for the Security Device. NOTE: Your Computer will reboot during restart in order to change State of Security Device.

Platform Hierarchy

Enable or Disable Platform Hierarchy

Storage Hierarchy

Enable or Disable Storage Hierarchy

Endorsement Hierarchy

Enable or Disable Endorsement Hierarchy

HashPolicy

Select the Hash policy to use. SHA-2 is most secure but might not be supported by all Operating Systems

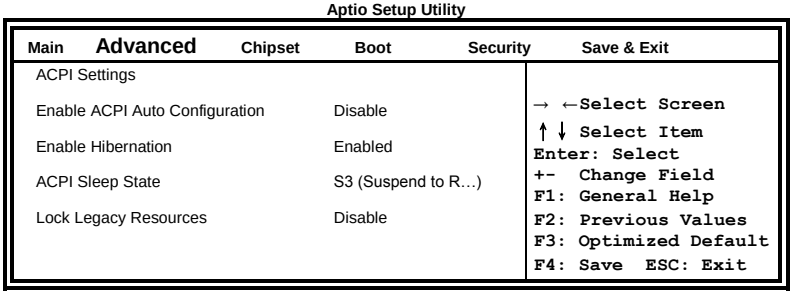
TPM 20 InterfaceType

Select the Communication Interface to TPM 20 Devices.

Device Select

TPM 1.2 will restrict support to TPM 1.2 devices, TPM 2.0 will restrict support to TPM 2.0 devices, Auto will support both with the default set to TPM 2.0 devices if not found, TPM 1.2 devices will be enumerated.

ACPI Settings



Enable ACPI Auto Configuration

Enables or Disables BIOS ACPI Auto Configuration.

Enable Hibernation

Enables or Disables System ability to Hibernate (OS/S4 Sleep State). This option may be not effective with some OS.

ACPI Sleep State

Select the highest ACPI sleep state the system will enter when the SUSPEND button is pressed.

Lock Legacy Resources

Enables or Disables Lock of Legacy Resources

LVDS (eDP/DP) Configuration

Aptio Setup Utility

Main	Advanced	Chipset	Boot	Security	Save & Exit
LVDS (eDP/DP) Configuration					
LVDS (eDP/DP) Support			Disabled		

→ ←Select Screen
↑ ↓ Select Item
Enter: Select
+- Change Field
F1: General Help
F2: Previous Values
F3: Optimized Default
F4: Save ESC: Exit

LVDS (eDP/DP) Support

LVDS (eDP/DP) ON/OFF

iSmart Controller

Aptio Setup Utility

Main	Advanced	Chipset	Boot	Security	Save & Exit
iSmart Controller					
Power-On after Power failure			Disable		
Temperature Guardian			Disable		
Schedule Slot 1			None		
Schedule Slot 2			None		

→ ←Select Screen
↑ ↓ Select Item
Enter: Select
+- Change Opt.
F1: General Help
F2: Previous Values
F3: Optimized Defaults
F4: Save & Exit
ESC: Exit

Power-On after Power failure

This field sets the system power status whether Disable or Enable when power returns to the system from a power failure situation.

Temperature Guardian

Generate the reset signal when system hangs up on POST.

Schedule Slot 1 / 2

Setup the hour/minute for system power on.

NCT5523D Super IO Configuration

Aptio Setup Utility

Main	Advanced	Chipset	Boot	Security	Save & Exit
NCT5523D Super IO Configuration					
Super IO Chip			NCT5523D		
► Serial Port 1 Configuration					
► Serial Port 2 Configuration					

→ ←Select Screen
↑ ↓ Select Item
Enter: Select
+- Change Field
F1: General Help
F2: Previous Values
F3: Optimized Default
F4: Save
ESC: Exit

Serial Port Configuration

Set Parameters of Serial Ports (COM)

Serial Port 1 Device Mode:

(1) RS232,
(2) RS485, (3) RS485 with termination,
(4) RS422, (5) RS422 with termination

NCT5523D H/W Monitor

Aptio Setup Utility

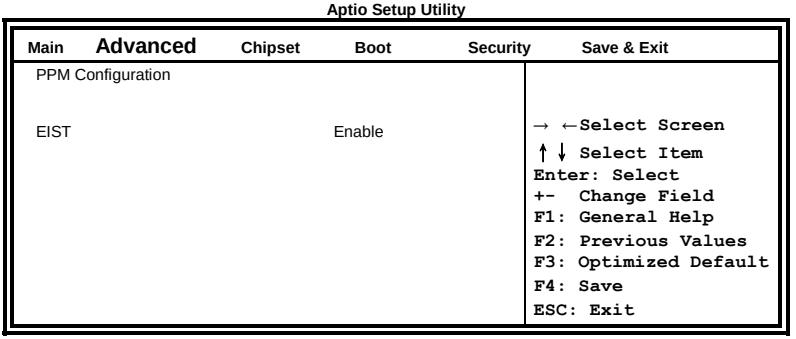
Main	Advanced	Chipset	Boot	Security	Save & Exit
PC Health Status					
ACPI Shutdown Temperature			Disable		
System	Temperature		+39.0 C		
CPU	Temperature		+39.5 C		
Vcore			+0.872 V		
VCC3V			+3.360 V		
3VSB			+3.120V		

→ ←Select Screen
↑ ↓ Select Item
Enter: Select
+- Change Field
F1: General Help
F2: Previous Values
F3: Optimized Default
F4: Save
ESC: Exit

Temperatures/Voltages

These fields are the parameters of the hardware monitoring function feature of the board. The values are read-only values as monitored by the system and show the PC health status.

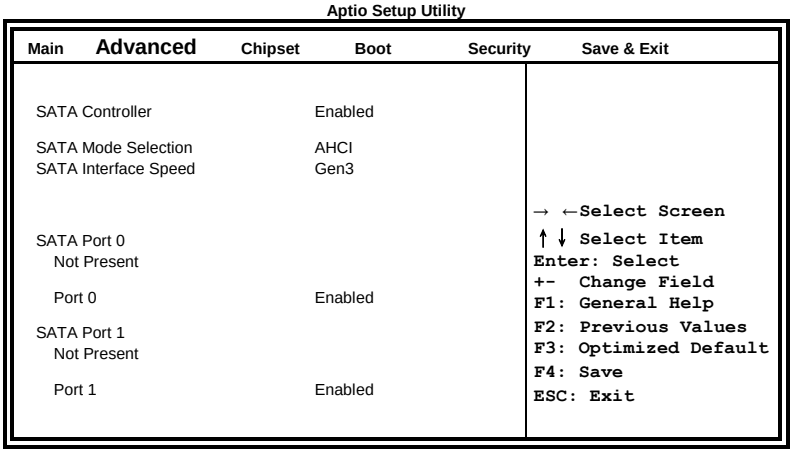
PPM Configuration



EIST

Enaable / Disable Intel SpeedStep

SATA Configuration



SATA Controller
Enable / Disable SATA Device

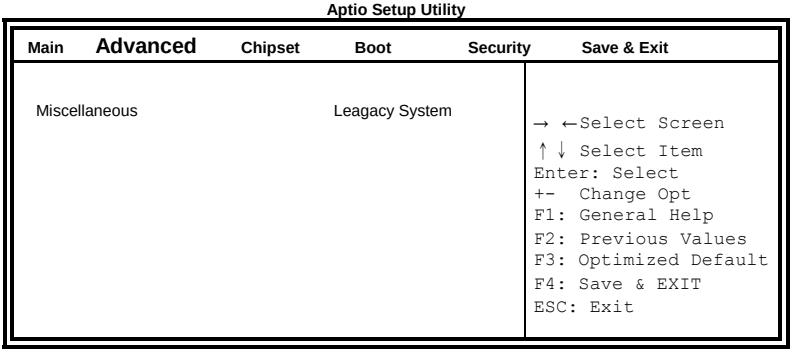
SATA Mode Selection
Determines how SATA controller operates.

SATA Interface Speed
Select SATA Interface Speed CHV A1 always with Gen1 Speed.

Port
Enable or Disable SATA Port

Hot Plug
Designates this port as Hot Pluggable.

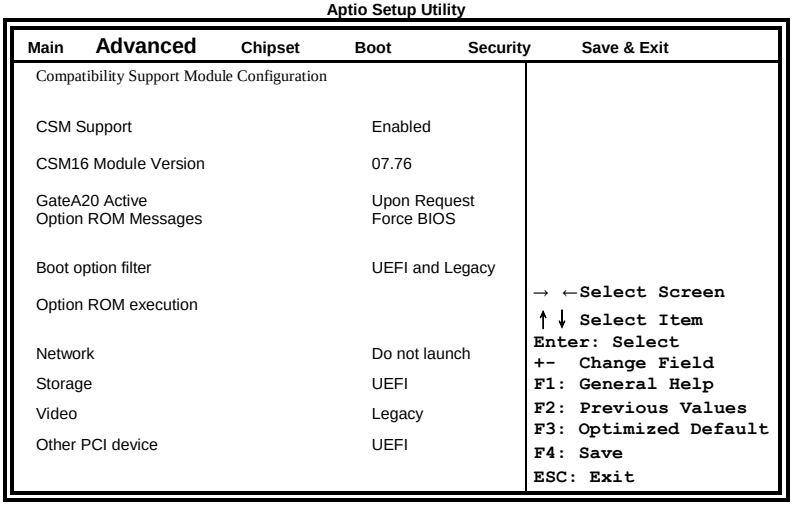
Miscellaneous Configuration



Miscellaneous

BOM Config

CSM Configuration



CSM Support
Enable/Disable CSM Support.

Boot option filter
This option controls Legacy/UEFI ROMs priority

Network
Controls the execution of UEFI and Legacy PXE OpROM

Storage
Controls the execution of UEFI and Legacy Storage OpROM

Video
Controls the execution of UEFI and Legacy Video OpROM

Other PCI device
Determines OpROM execution policy for devices other than Network, Storage, or Video.

USB Configuration

Aptio Setup Utility

Main	Advanced	Chipset	Boot	Security	Save & Exit
USB Configuration					
USB Module Version			11		
USB Controllers:					
1XHCI					
USB Devices:					
1 Drive, 1 Keyboard, 1 Hubs					
Legacy USB Support			Enabled		
XHCI Hand-off			Enabled		→ ←Select Screen
USB Mass Storage Driver Support			Enabled		↑↓ Select Item
					Enter: Select
USB hardware delays and time-outs:					+- Change Field
USB Transfer time-out			20 sec		F1: General Help
Device reset tine-out			20 sec		F2: Previous Values
Device power-up delay			Auto		F3: Optimized Default
					F4: Save
					ESC: Exit

Legacy USB Support

Enables Legacy USB support.
AUTO option disables legacy support if no USB devices are connected.
DISABLE option keeps USB devices available only for EFI applications.

XHCI Hand-off

This is a workaround for OSes without XHCI hand-off support. The XHCI ownership change should be claimed by XHCI driver.

USB Mass Storage Driver Support

Enable/Disable USB Mass Storage Driver Support.

USB Transfer time-out

The time-out value for Control, Bulk, and Interrupt transfers.

Device reset time-out

USB mass Storage device start Unit command time-out.

Device power-up delay

Maximum time the device will take before it properly reports itself to the Host Controller. 'Auto' uses default value: for a Root port it is 100ms, for a Hub port the delay is taken from Hub descriptor.

Chipset Settings

Aptio Setup Utility

Main	Advanced	Chipset	Boot	Security	Save & Exit
► North Bridge ► South Bridge					

North Bridge

Aptio Setup Utility

Main	Advanced	Chipset	Boot	Security	Save & Exit
Memory Information					
Total Memory			8192 MB(LPDDR3)		
Memory Slot 0			4096 MB(LPDDR3)		
Memory Slot 1			4096 MB(LPDDR3)		
Max TOLUD			Dynamtic		

→ ←Select Screen
↑ ↓ Select Item
Enter: Select
+- Change Field
F1: General Help
F2: Previous Values
F3: Optimized Default
F4: Save ESC: Exit

South Bridge

Aptio Setup Utility

Main	Advanced	Chipset	Boot	Security	Save & Exit
► Security Configuration					
► Azalia Configuration					
Restore AC Power Loss			Power off		

Azalia Configuration
Azalia HD Audio Options

USB Configuration
USB Configuration Settings

PCI Express Configuration
PCI Express Configuration Settings

Azalia Configuration

Main	Advanced	Chipset	Boot	Security	Save & Exit
Azalia Configuration					
Audio Controller			Enabled		

.

Audio Controller

Control Detection of the Azalia device.
Disable= Azalia will be unconditionally disabled.
Enable= Azalia will be unconditionally Enable.

Security Settings

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

Aptio Setup Utility

Main	Advanced	Chipset	Boot	Security	Save & Exit
Password Description					
If ONLY the Administrator's password is set, then this only limit access to Setup and is only asked for when entering Setup. If ONLY the User's password is set, then this is a power on password and must be entered to boot or enter Setup. In Setup the User will have Administrator rights					
The password length must be					→ ←Select Screen
in the following range:					↑ ↓ Select Item
Minimum length			3		Enter: Select
Maximum length			20		+- Change Field
					F1: General Help
					F2: Previous Values
					F3: Optimized Default
Administrator Password					F4: Save
User Password					ESC: Exit

Administrator Password

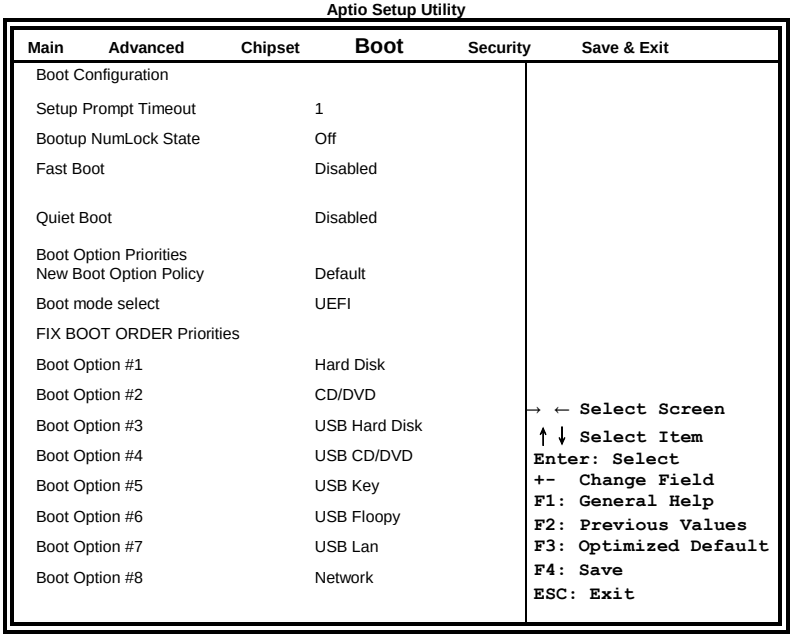
Set Setup Administrator Password.

User Password

Set User Password.

Boot Settings

This section allows you to configure the boot settings.



Setup Prompt Timeout

Number of seconds to wait for setup activation key.
65535(0xFFFF) means indefinite waiting.

Bootup NumLock State

Select 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 active boot option. Has no effect for BBS boot options.

Save & Exit Settings

Aptio Setup Utility

Main Advanced Chipset Boot Security	Save & Exit
Reset System with ME disable ModeMEUD000	
Save Options	
Save Changes and Exit	
Discard Changes and Exit	
Save Changes and Reset	
Discard Changes and Reset	
Save Changes	
Discard Changes	→ ←Select Screen
	↑ ↓ Select Item
Defaults Options	Enter: Select
Restore Defaults	+- Change Field
Save as User Defaults	F1: General Help
Restore User Defaults	F2: Previous Values
	F3: Optimized Default
	F4: Save
	ESC: Exit

Save Changes and Exit
Exit system setup after saving the changes.

Discard Changes and Exit
Exit system setup without saving any changes.

Save Changes and Reset
Reset the system after saving the changes.

Discard Changes and Reset
Reset system setup without saving any changes.

Save Changes
Save Changes done so far to any of the setup options.

Discard Changes
Discard Changes done so far to any of the setup options.

Restore Defaults
Restore/Load Defaults values for all the setup options.

Save as User Defaults
Save the changes done so far as User Defaults.

Restore User Defaults
Restore the User Defaults to all the setup options.

Drivers Installation

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

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

Intel Chipset Software Installation Utility

The Intel Chipset Drivers should be installed first before the software drivers to enable Plug & Play INF support for Intel chipset components. Follow the instructions below to complete the installation.

1. Insert the disc that comes with the board. Click ***Intel*** and then ***Intel(R) Braswell Chipset Drivers.***

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. Click ***Accept*** to accept the software license agreement and proceed with the installation process.

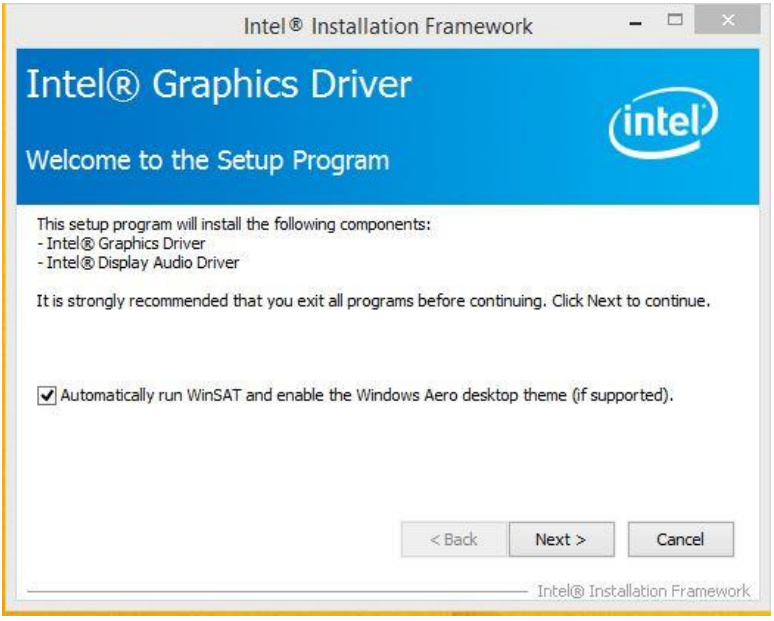
5. On the Readme File Information screen, click ***Install*** to continue the installation.

6. The Setup process is now complete. Click ***Finish*** to restart the computer and for changes to take effect.

VGA Drivers Installation

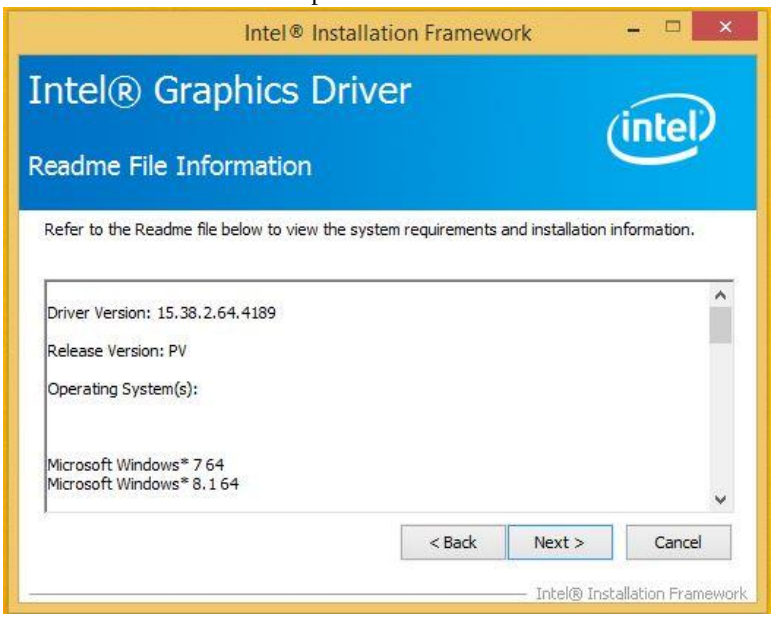
1. Click ***Intel(R) Braswell Graphics Driver***

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



3. Click ***Yes*** to to agree with the license agreement and continue the installation.

4. On the Readme File Information screen, click ***Next*** to continue the installation of the Intel® Graphics Driver.



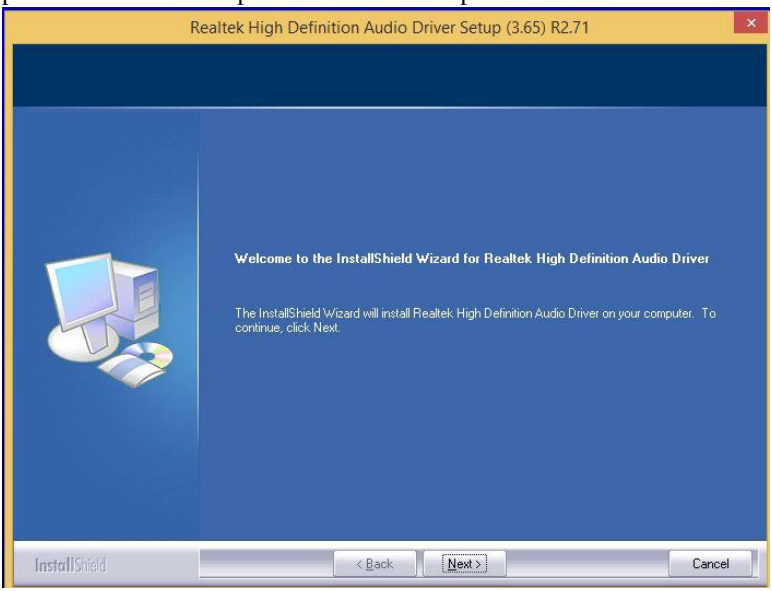
5. On Setup Progress screen, click ***Next*** to continue.

6. Setup complete. Click ***Finish*** to restart the computer and for changes to take effect.

Realtek HD Audio Driver Installation

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

2. On the Welcome to the InstallShield Wizard screen, click ***Next*** to proceed with and complete the installation process.



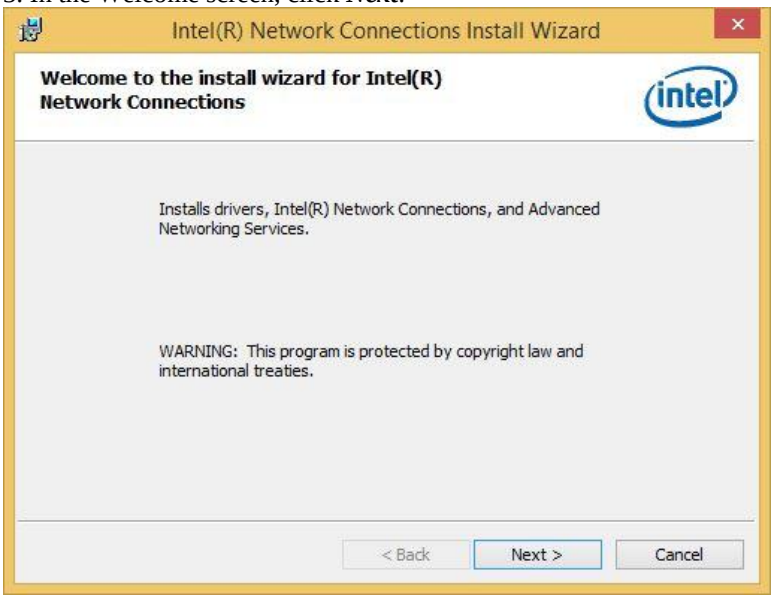
3. Restart the computer when prompted.

LAN Drivers Installation

1. Insert the CD that comes with the board. Click ***LAN Card*** and then ***Intel LAN Controller Drivers***.

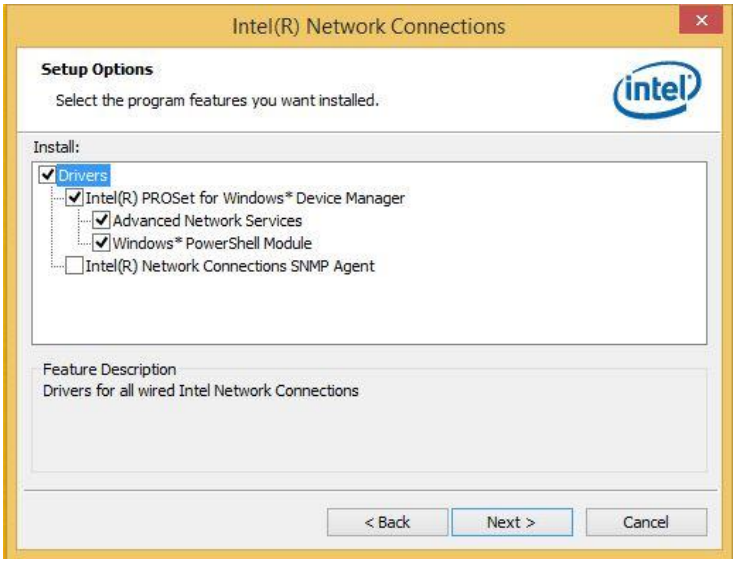
2. Click ***Intel(R) I21x Gigabit Network Drivers***

3. In the Welcome screen, click ***Next***.

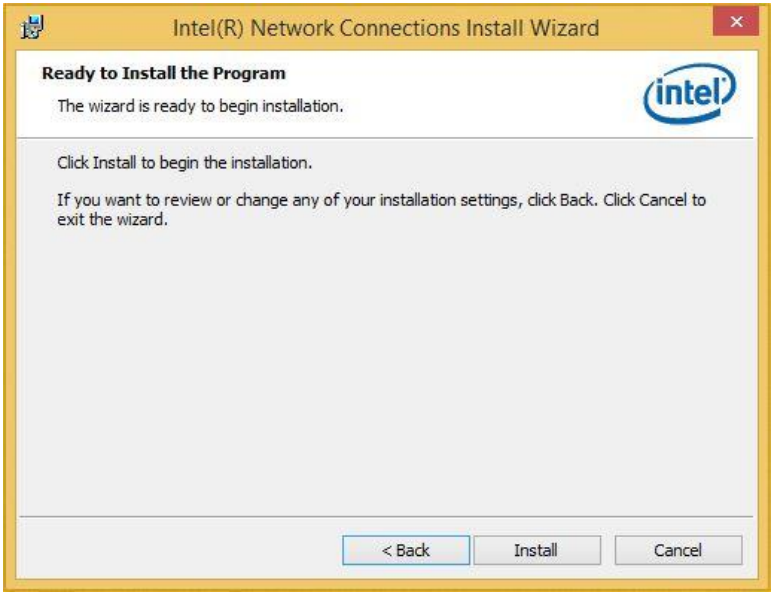


4. In the License Agreement screen, click ***I accept the terms in license agreement*** and ***Next*** to accept the software license agreement and proceed with the installation process.

5. Click the checkbox for **Drivers** in the Setup Options screen to select it and click **Next** to continue.



6. When the Ready to Install the Program screen appears, click ***Install*** to continue.

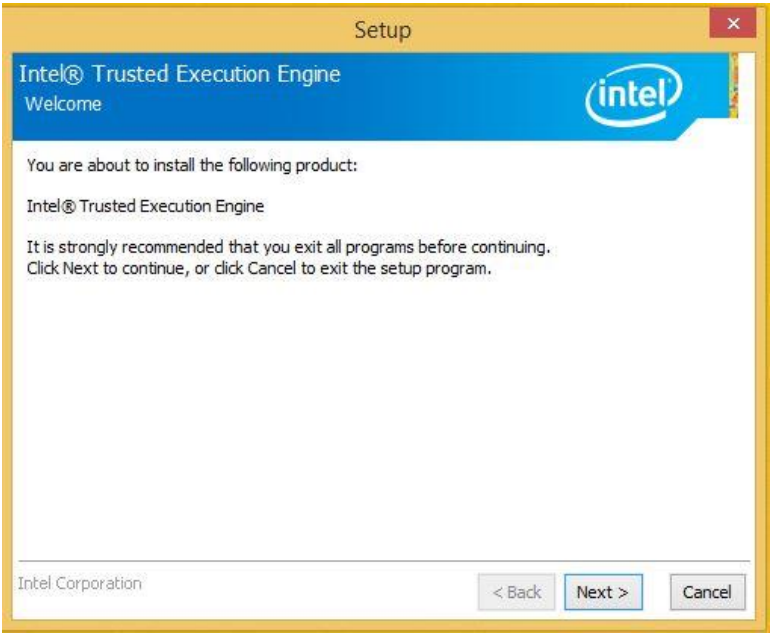


7. When InstallShield Wizard is complete, click ***Finish***.

TXE Drivers Installation

1. Click ***Intel(R) TXE Driver***

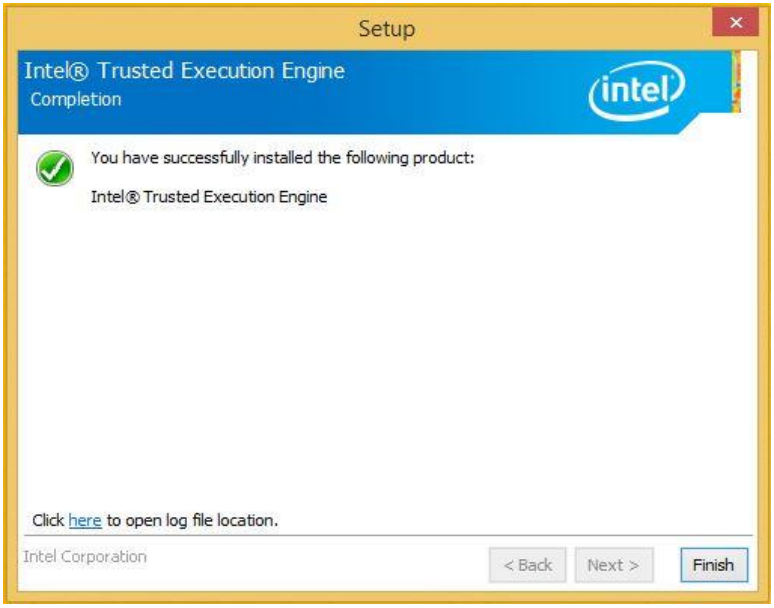
2. When the Welcome screen appears, click ***Next*** to continue.



3. Click ***Next*** to to agree with the license agreement and continue the installation.

4. On the Confirmation screen, click ***Next*** to continue the installation of the Intel(R) TXE Driver.

5. Setup complete. Click ***Finish*** to restart the computer and for changes to take effect.



Appendix

A. I/O Port Address Map

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

Address	Device Description
000h - FFFh	PCI Express Standard Root Port
040h - 043h	System timer
070h - 077h	System CMOS/real time clock
D00h – FFFh	PCI Bus
2F8h – 2FFh	Communications Port (COM2)
3B0h – 3BBh	Intel(R) HD Graphics
3F8h – 3FFh	Communications Port (COM1)
040h – 05Fh	Intel(R) Celeron(R)/Pentium(R) SM Bus Controller
060h – 07Fh	Standard SATA AHCI Controller

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
IRQ0	System timer
IRQ3	Communications Port (COM2)
IRQ4	Communications Port (COM1)
IRQ10	Intel(R) Celeron(R)/Pentium(R) SM Bus Controller
IRQ 18	Intel(R) Atom(TM)/Celeron(R)/Pentium(R) Processor SD Host Controller
IRQ19	Standard SATA AHCI Controller
IRQ22	High Definition Audio Controller

C. Watchdog Timer Configuration

The 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, the user will restart the WDT at regular intervals before the timer counts to zero.

SAMPLE CODE:

File of the 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
```

File of the 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
}
//-------------------------------------------------------------------------
```

File of the 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. Digital I/O Sample Code

File of the 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
```

File of the 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);
}
//---------------------------------------------------------------------------
```

File of the 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;
}
//----------------------------------------------------------------------------
```