IB903F

AMD® Embedded G-Series
3.5” Disk Size SBC

USER’S MANUAL

Version 1.0

Acknowledgments

Table of Contents

This page is intentionally left blank.

Introduction

Product Description

The IB903F is a 3.5-inch single board computer based on the AMD® Embedded G-series (SoC) APU. The Embedded G-series (SoC) APU -based board delivers an exceptionally high-definition visual experience and offers improved CPU performance over Embedded G-series (SoC) APU. The SOC's advanced GPU enables parallel processing and high-performance graphics processing that provides up to 20% improvement vs. AMD G-Series APU.

The IB903F platform is well-suited for low-power and high-performance designs in a broad range of markets including Industrial Control & Automation, Digital Signage, Thin Client, Electronic Gaming Machines, and SMB storage appliances.

IB903F Features:

- Supports AMD® Embedded G-series (SoC) processors
- One DDR3 SO-DIMM, 1333/1600 MHz, Max. 8GB memory
- Integrated graphics for DVI-I, LVDS displays
- 2 x SATA III connector
- 4x COM port connector
- 2 x Mini-PCIe(x1) slot *(w/ USB support)*
- 2x GbE (RJ-45) connector
- 1x 9V to 24V DC-IN power connector

Checklist

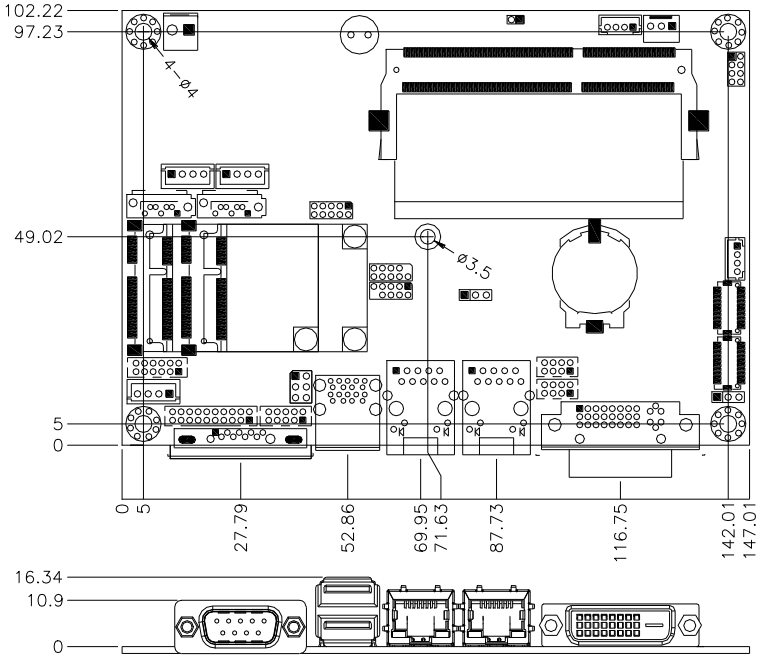
Your IB903F package should include the items listed below.

- The IB903F SBC
- This User's Manual
- 1 CD containing chipset drivers and flash memory utility

IB903F Specifications

Product Name	IB903F (Default model name on PCB surface)
Form Factor	3.5" Disk Size SBC
CPU Type/Speed	AMD Embedded G-series SoC (CPU+GPU+NB), 28nm process technology Quad Core @ 2.0 GHz (TDP=25 W) [Default, IB903F-Q2G] Dual Core @ 1.6 GHz (TDP=15W) [Optioanl][IB903F-16] Dual Core @ 1.5 GHz (TDP=9W) [Optioanl][IB903F]
Chipset	FT3 BGA package (24.5mm x 24.5mm)
BIOS	AMI BIOS, supports ACPI Function
Memory	1 x DDR3(L)-1600 @ 1.5V(1.35V) SO-DIMM, Single Channel (Max. 8GB) ,No-ECC
VGA	AMD Embedded G-series SoC built-in GPU, supports 2 independent displays, DirectX® 11.1, DisplayPort 1.2, - DVI-I (DP#1[DVI] + CRT)
LVDS	24-bit dual-channel LVDS – via SoC DP#0 w/ Analogix ANX3110
LAN	Realtek RTL8111G-CG x 2 for dual PCIe GbE LAN 32-pin QFN package
USB	2 x USB 3.0 Ports – via SoC built-in XHCIs 6 x USB 2.0 Ports – via SoC built-in EHCIs
Serial ATA Ports	SoC Integral SATA III (6.0Gb/s) controllers, supports 2 ports
Audio	SoC Integral HDA + Realtek ALC269Q-VC2 Codec [6mm x 6mm @ MQFN48] with class-D speaker amplifier(2.3W per channel @ 5V power supply) Supportf 2-channel audio out + amp
LPC I/O	Fintek F81846AD-I - COM#1(RS232/422/485), jumper-less select(SP339), supports pin-9 with power(500mA) - COM#2 ~#4 (RS232 only) - Hardware Monitor (2 thermal inputs, 4 voltage monitor inputs & 1 x Fan header)
Digital IO	4 in & 4 out
Expansion Slots	Mini PCI-e socket x 2 (1x Full-sized+1xHalf-sized) **Full-sized MiniPCIe(1x) supports mSATA**
Edge Connector	DVI-I, 2x RJ45, 2x USB, COM1
Onboard Header/ Connector	4x USB 2.0, 2x DF13 for 24-bit dual channel LVDS, Audio, speaker, COM2, COM3/4, LPC (80-port card debugging purpose), 2x Mini PCI-e(1x), backlight/brightness control, 2x power connector SATA HDD [JST type]
Watchdog Timer	Yes (256 segments, 0, 1, 2…255 sec/min)
Power Connector	+9V ~ +24V DC-in
Others	- i-Smart function (TI MSP430G2433 MCU) - AT24C02C EEPROM [SO8 type] via SMbus (Optional)
OS Support	- Windows 7, Windows 8 - Linux
RoHS	Yes
Board Size	102mm x 147mm

Board Dimensions



Installations

This section provides information on how to use the jumpers and connectors on the IB903F in order to set up a workable system. The topics covered are:

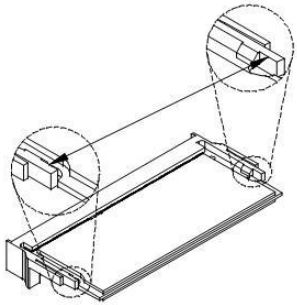
Installing the Memory

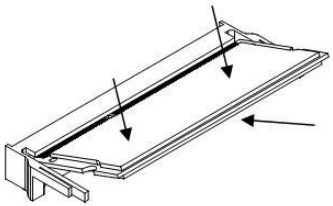
The IB903F board supports one DDR3 memory socket for a maximum total memory of 8GB DDR3 memory type.

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.

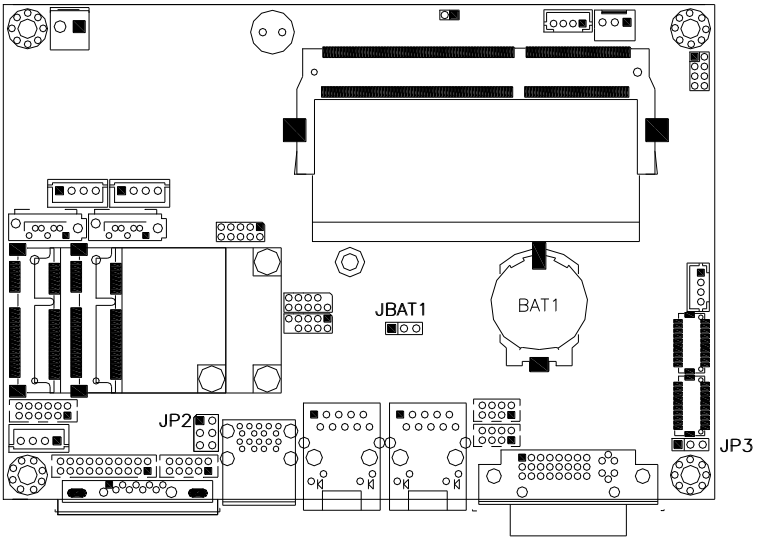




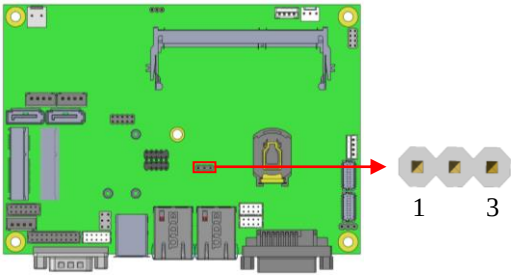
Setting the Jumpers

Jumpers are used on IB903F 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 IB903F and their respective functions.

Jumper Locations on IB903F

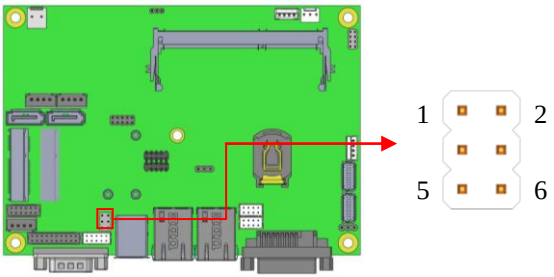


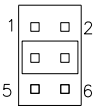
JBAT1: Clear CMOS Contents



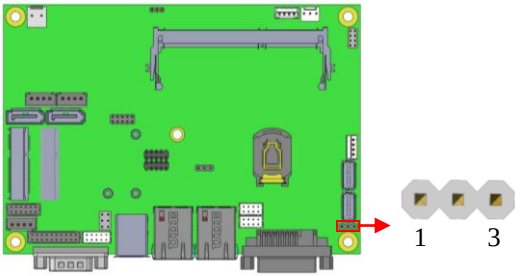
JBAT1	Setting	Function
	Pin 1-2 Short/Closed	Normal
	Pin 2-3 Short/Closed	Clear CMOS

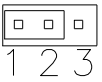
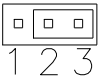
JP2: COM1 RS232 RI/+5V/+12V Power Setting



JP2	Setting	Function
	Pin 1-3 Short/Closed	+12V
	Pin 3-4 Short/Closed	RI
	Pin 3-5 Short/Closed	+5V

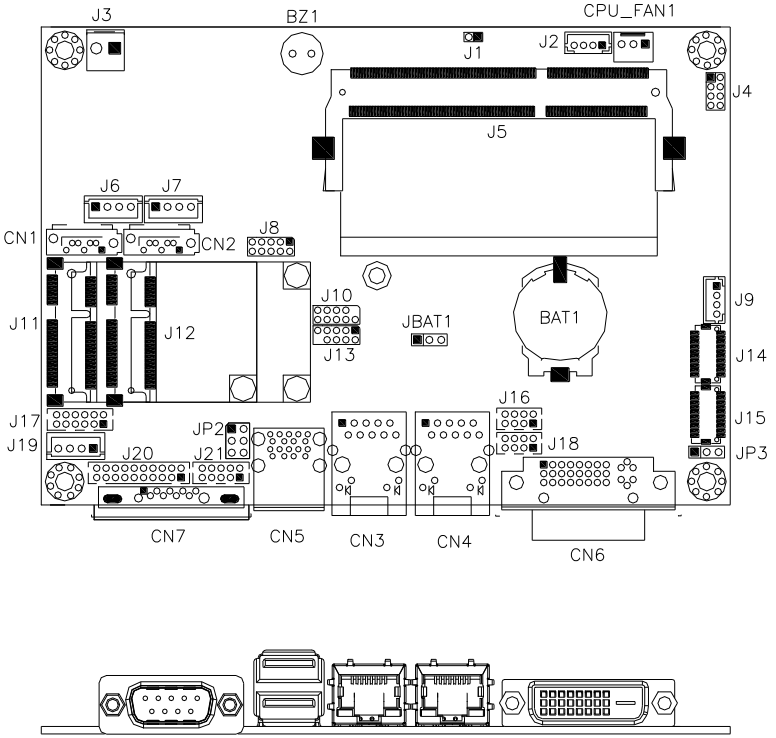
JP3: LVDS Panel Power Selection

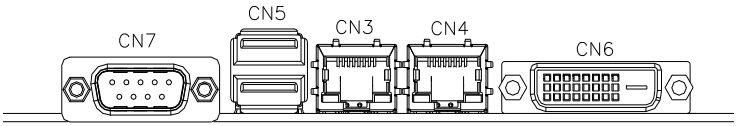


JP3	Setting	Panel Voltage
	Pin 1-2 Short/Closed	3.3V (default)
	Pin 2-3 Short/Closed	5V

Connectors on IB903F

Connector Locations on IB903F





CN3, CN4: Gigabit LAN (RTL8111G)

CN4 RJ45 LAN connector (only) features for EuP LAN wakeup.

CN5: USB 1/2 Connector

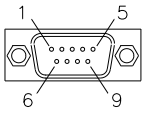
CN6: VGA DVI-I Connector

CN7: DB9 Connector

(COM1) is a DB-9 connector.

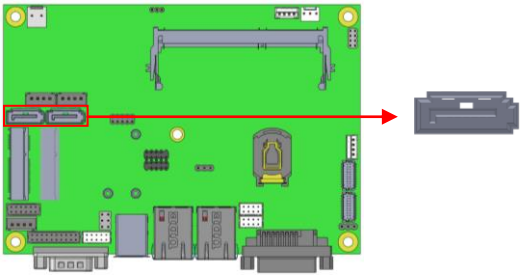
Signal Name	Pin #	Pin #	Signal Name
DCD, Data carrier detect	1	6	DSR, Data set ready
RXD, Receive data	2	7	RTS, Request to send
TXD, Transmit data	3	8	CTS, Clear to send
DTR, Data terminal ready	4	9	RI, Ring indicator
GND, ground	5	10	Not Used

COM1 is jumper-less for RS-232, RS-422 and RS-485 and is to be configured with BIOS Selection.

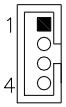


Pin #	Signal Name		
	RS-232	R2-422	RS-485
1	DCD	TX-	DATA-
2	RX	TX+	DATA+
3	TX	RX+	NC
4	DTR	RX-	NC
5	Ground	Ground	Ground
6	DSR	NC	NC
7	RTS	NC	NC
8	CTS	NC	NC
9	RI	NC	NC
10	NC	NC	NC

CN1, CN2: SATA Connectors

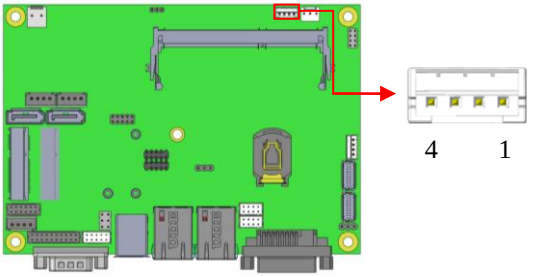


J9: LCD Backlight Connector

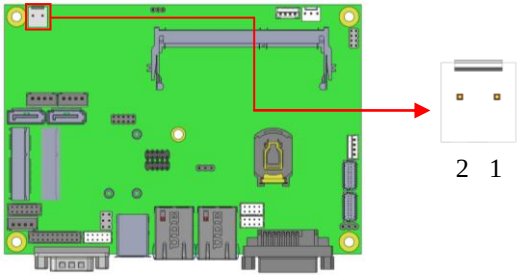


Pin #	Signal Name
1	Backlight Power
2	Backlight Enable
3	Brightness Control
4	Ground

J2: MCU Flash Connector (factory use only)



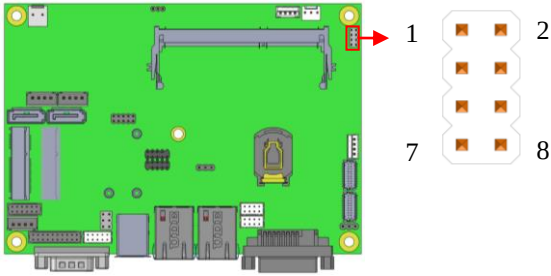
J3: Board Input Power Connector



Pin #	Signal Name
1	+9V to +24V
2	GND

J4: Front Panel Connector

The following table shows the pin outs of the 2x4 pin header



Signal Name	Pin #	Pin #	Signal Name
Ground	1	2	PWR_SW
PWR_LED+	3	4	PWR_LED-
HDD_LED+	5	6	HDD_LED-
Ground	7	8	RESET

J4 provides connectors for system indicators that provide light indication of the computer activities and switches to change the computer status. J4 is an 8-pin header that provides interfaces for the following functions.

ATX Power ON Switch: Pins 1 and 2
This 2-pin connector is an "ATX Power Supply On/Off Switch" on the system that connects to the power switch on the case. When pressed, the power switch will force the system to power on. When pressed again, it will force the system to power off.

Power LED: Pins 3 and 4

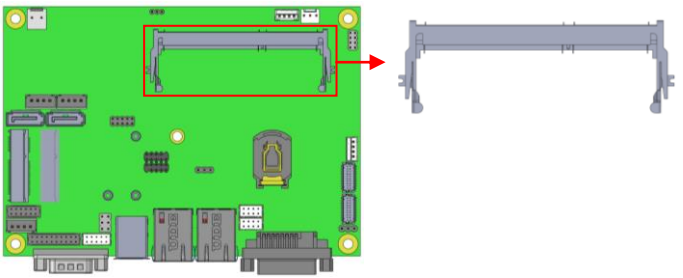
Pin #	Signal Name
3	LED(+)
4	LED(-)

Hard Disk Drive LED Connector: Pins 5 and 6
This connector connects to the hard drive activity LED on control panel. This LED will flash when the HDD is being accessed.

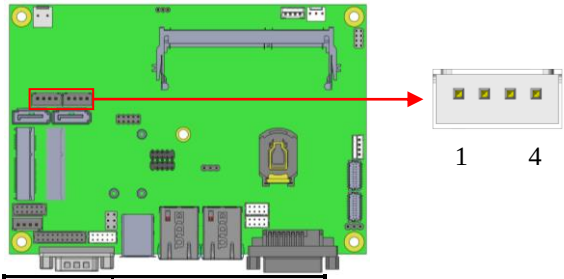
Pin #	Signal Name
5	LED(+)
6	LED(-)

Reset Switch: Pins 7 and 8
The reset switch allows the user to reset the system without turning the main power switch off and then on again. Orientation is not required when making a connection to this header.

J5: DDR3 SO-DIMM Socket

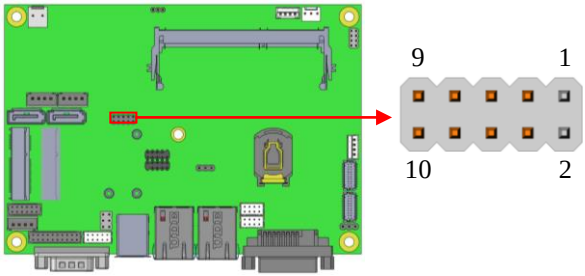


J6, J7: SATA HDD Power Connectors



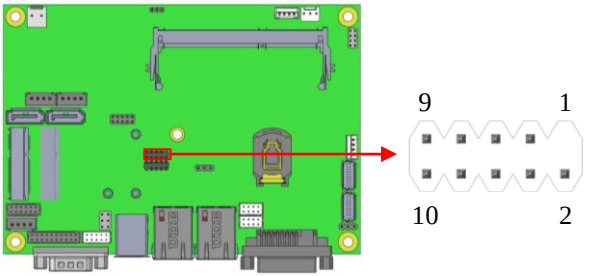
Pin #	Signal Name
1	+5V
2	Ground
3	Ground
4	+12V

J8: Digital I/O Connector

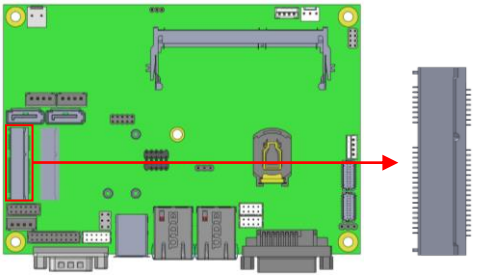


Signal Name	Pin	Pin	Signal Name
GND	1	2	VCC
OUT3	3	4	OUT1
OUT2	5	6	OUT0
IN3	7	8	IN1
IN2	9	10	IN0

J10: SPI Flash Connector (factory use only)

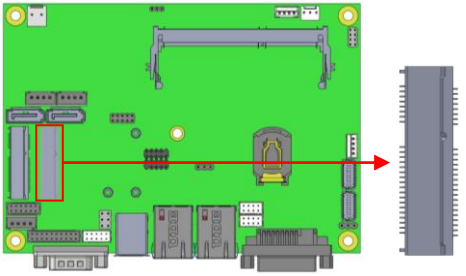


J11: Mini PCIE Connector (Supports mSATA)

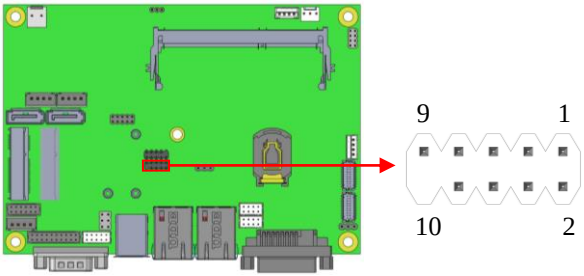


J11 also supports mSATA. However, when J11 is used for mSATA, then CN1 SATA port cannot be used. Only one of them can be used at one time to support SATA.

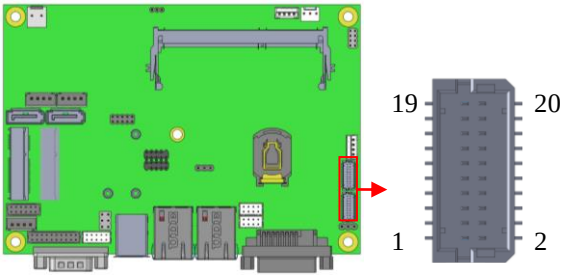
J12: Mini PCIE Connector (Half Size)



J13: Debug 80 Port Connector (factory use only)

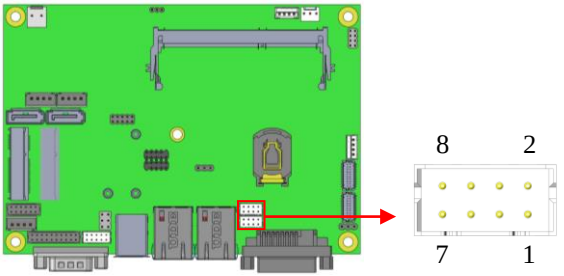


J14, J15: LVDS Connectors



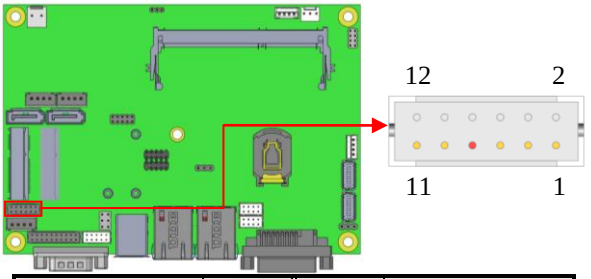
Signal Name	Pin #	Pin #	Signal Name
LCD_PWR	19	20	LCD_PWR
LD3+	17	18	LD3-
GND	15	16	GND
CLK+	13	14	CLK-
GND	11	12	GND
LD2+	9	10	LD2-
GND	7	8	GND
LD1+	5	6	LD1-
GND	3	4	GND
LD0+	1	2	LD0-

J16, J18: USB3/4/5/6 Connector



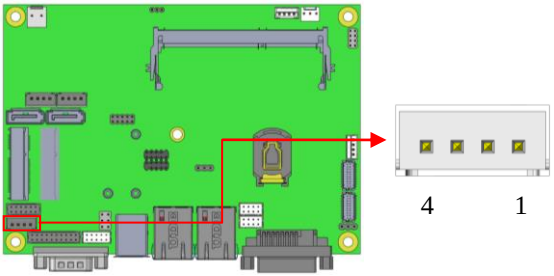
Signal Name	Pin #	Pin #	Signal Name
Vcc	1	2	Ground
D-	3	4	D+
D+	5	6	D-
Ground	7	8	Vcc

J17: Audio Connector (DF11 Connector)



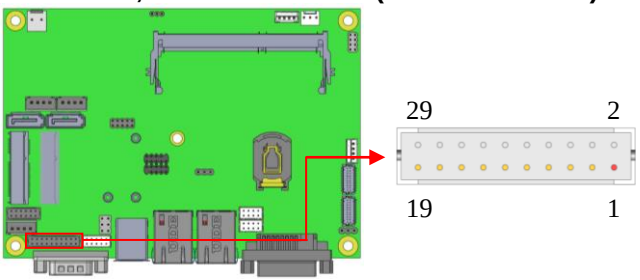
Signal Name	Pin #	Pin #	Signal Name
LINEOUT_R	2	1	LINEOUT_L
Ground	4	3	JD_FRONT
LINEIN_R	6	5	LINEIN_L
Ground	8	7	JD_LINEIN
MIC-R	10	9	MIC_L
Ground	12	11	JD_MIC1

J19: Amplify Connector



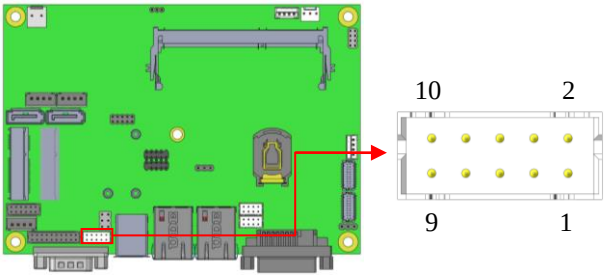
Pin #	Signal Name
1	OUTL+
2	OUTL-
3	OUTR-
4	OUTR+

J20: COM3, COM4 Serial Port (DF11 Connector)



Signal Name	Pin #	Pin #	Signal Name
DSR3	2	1	DCD3
RTS3	4	3	RXD3
CTS3	6	5	TXD3
RI3	8	7	DTR3
NC	10	9	Ground
DSR4	12	11	DCD4
RTS4	14	13	RXD4
CTS4	16	15	TXD4
RI4	18	17	DTR4
NC	20	19	Ground

J21: COM2/RS232 Serial Port



Signal Name	Pin #	Pin #	Signal Name
DCD, Data carrier detect	1	2	RXD, Receive data
TXD, Transmit data	3	4	Data terminal ready
GND, ground	5	6	DSR, Data set ready
RTS, Request to send	7	8	CTS, Clear to send
RI, Ring indicator	9	10	Not Used

BIOS Setup

This chapter describes the different settings available in the 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 provides critical low-level support for a standard device such as disk drives, serial ports and parallel ports. It also adds virus and password protection as well as special support for detailed fine-tuning of the chipset controlling the entire system.

BIOS Setup

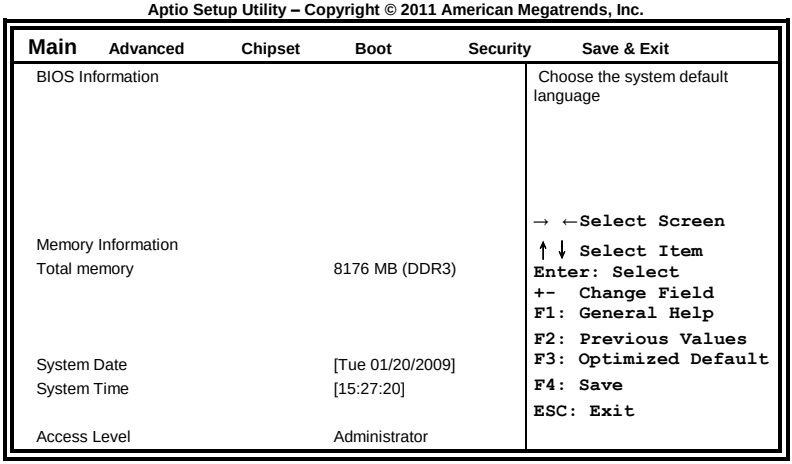
The BIOS provides a Setup utility program for specifying the system configurations and settings. The BIOS ROM of the system stores the Setup utility. When you turn on the computer, the BIOS is immediately activated. Pressing the <Del> key immediately allows you to enter the Setup utility. If you are 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> or <ESC>  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.

Main Settings



System Date

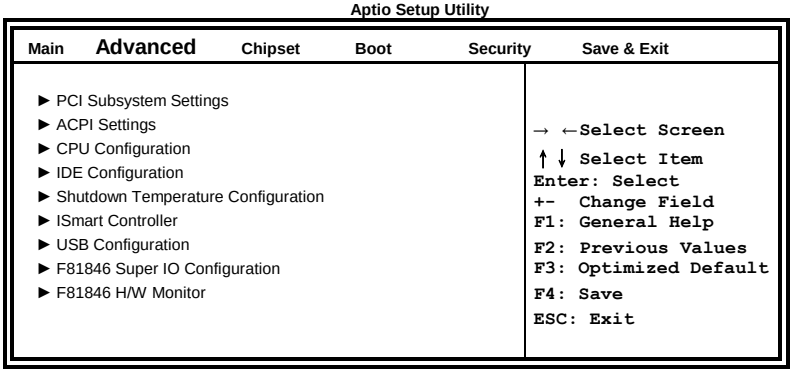
Set the Date. Use Tab to switch between Data elements.

System Time

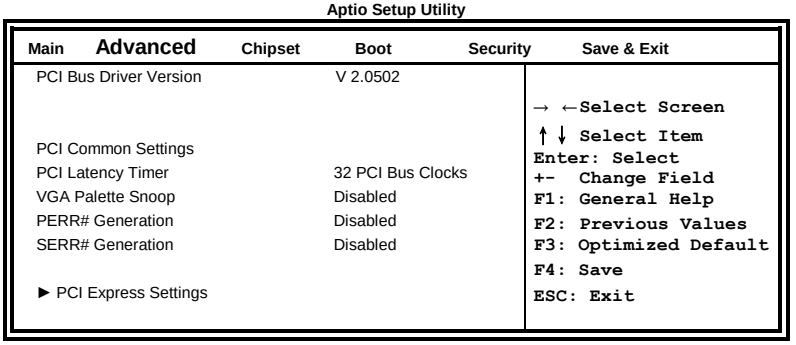
Set the Time. Use Tab to switch between Data elements.

Advanced Settings

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



PCI Subsystem Settings



PCI Latency Timer

Value to be programmed into PCI Latency Timer Register.

VGA Palette Snoop

Enables or disables VGA Palette Registers Snooping.

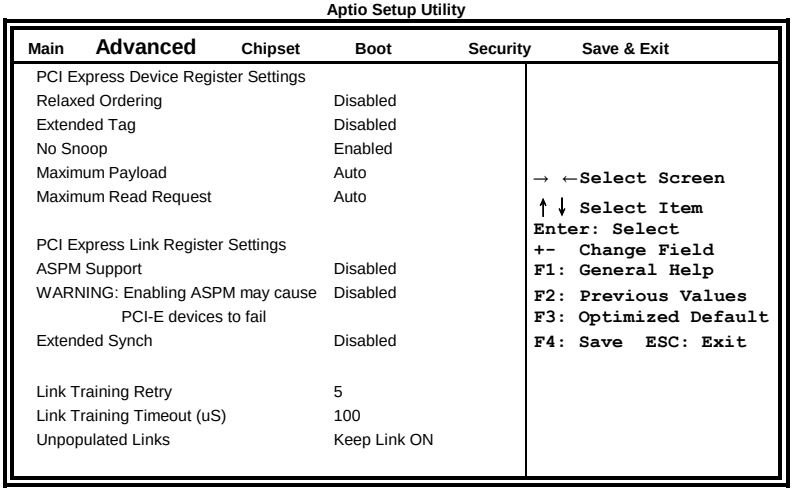
PERR# Generation

Enables or disables PCI device to generate PERR#.

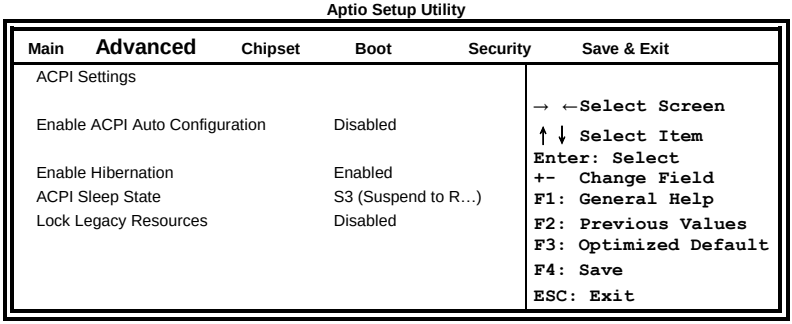
SERR# Generation

Enables or disables PCI device to generate SERR#.

PCI Express Settings



ACPI Settings



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

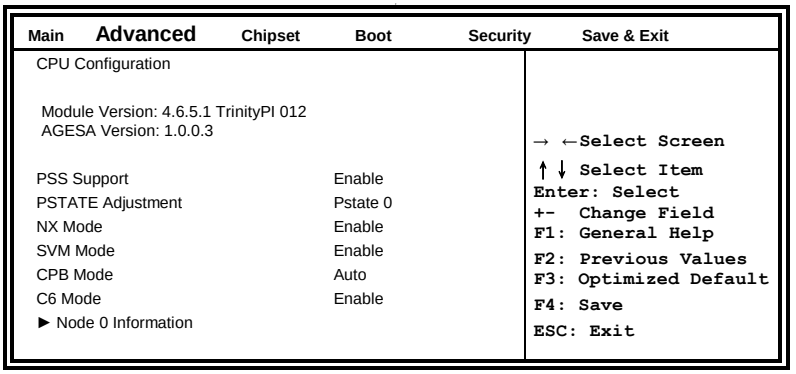
Select ACPI sleep state the system will enter, when the SUSPEND button is pressed.

Lock Legacy Resources

Enabled or Disabled Lock of Legacy Resources.

CPU Configuration

This section shows the CPU configuration parameters.



PSS Support

Enable/disable the generation of ACPI _PPC, _PPC, _PSS, and _PCT objects.

PSTATE Adjustment

Provide to adjust startup P-state level.

PPC Adjustment

Provide to adjust _PPC object.

NX Mode

Enable/disable No-execute page protection function.

SVM Mode

Enable/disable CPU Virtualization.

CPB Mode

Enable/disable CPB.

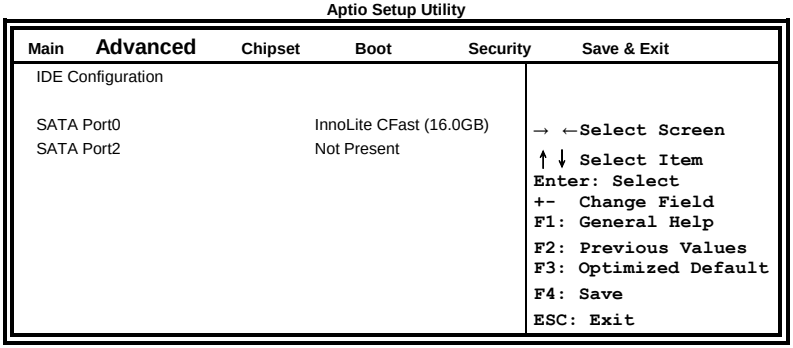
C6 Mode

Auto/disable CPB.

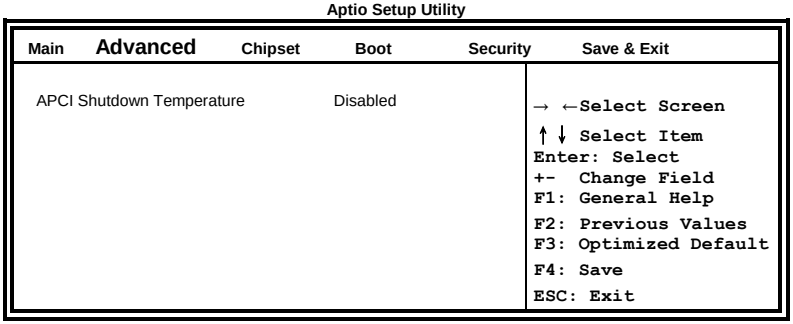
Node 0 Information

View memory information related to Node 0.

IDE Configuration



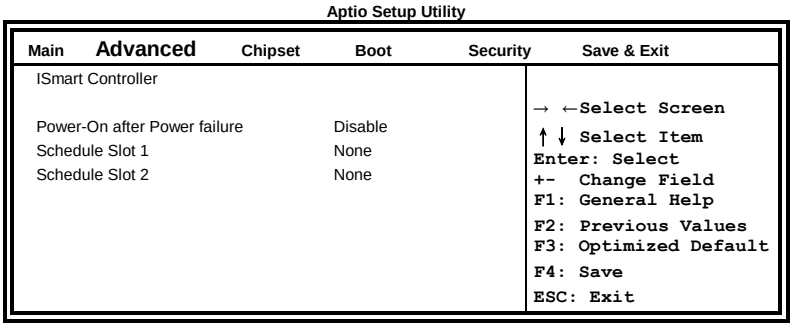
Shutdown Temperature Configuration



ACPI Shutdown Temperature

The default setting is Disabled.

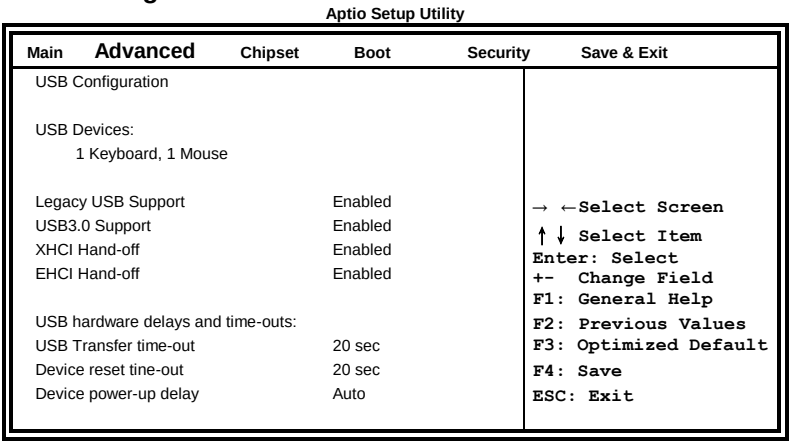
ISmart Controller



Power-On after Power failure
Enable or Disable.

Schedule Slot 1 / 2
Setup the hour/minute for system power on.

USB Configuration



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.

USB3.0 Support
Enable/Disable USB3.0 (XHCI) Controller support.

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

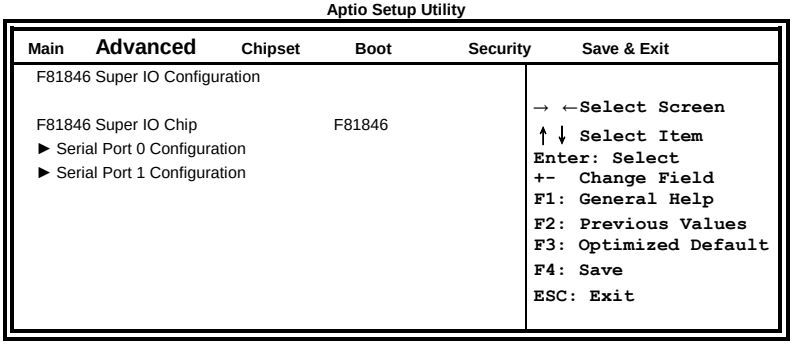
EHCI Hand-off
Enabled/Disabled. This is a workaround for OSes without EHCI hand-off support. The EHCI ownership change should be claimed by EHCI driver.

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.

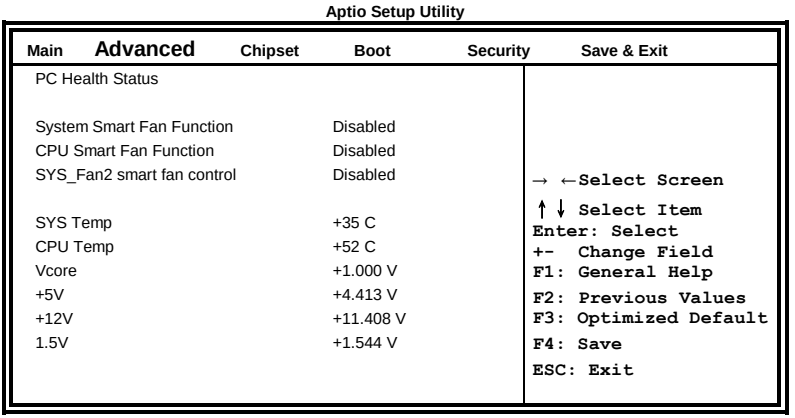
Device power-up delay
Maximum time the device will take before it properly reports itself to the Host Controller. 'Auto' uses default value: for a Root port it is 100ms, for a Hub port the delay is taken from Hub descriptor.

F81846 Super IO Configuration



Serial Port Configuration
Set Parameters of Serial Ports. User can Enable/Disable the serial port and Select an optimal settings for the Super IO Device.

F81846 H/W Monitor



Temperatures/Voltages

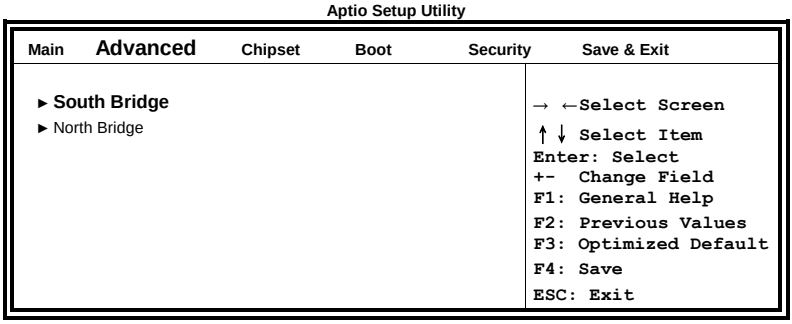
These fields are the parameters of the hardware monitoring function feature of the board. The values are read-only values as monitored by the system and show the PC health status.

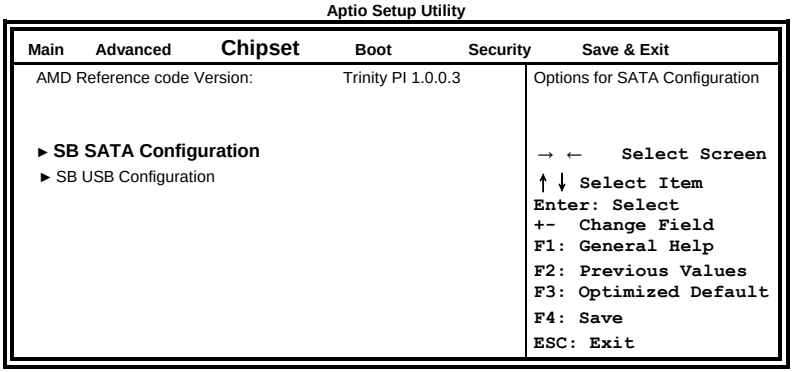
Smart Fan Function

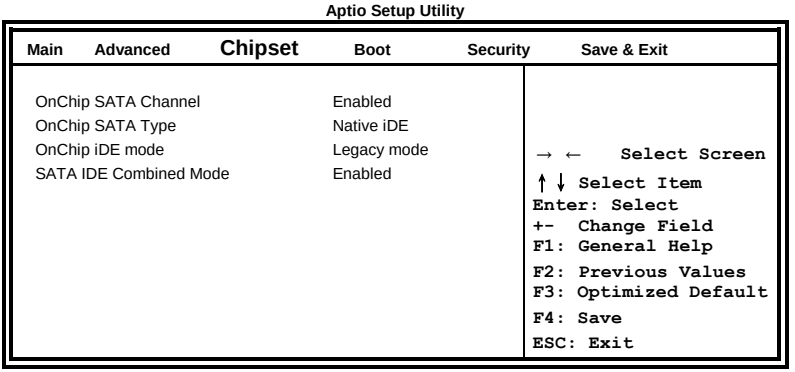
This field enables or disables the smart fan feature. At a certain temperature, the fan starts turning. Once the temperature drops to a certain level, it stops turning again.

Chipset Settings

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







OnChip SATA Channel
Enabled or Disabled.

OnChip SATA Type
Native IDE /n RAID /n AHCI /n AHCI /n Legacy IDE /n IDE->AHCI /n HyperFlash

OnChip IDE mode
Legacy mode or Native mode

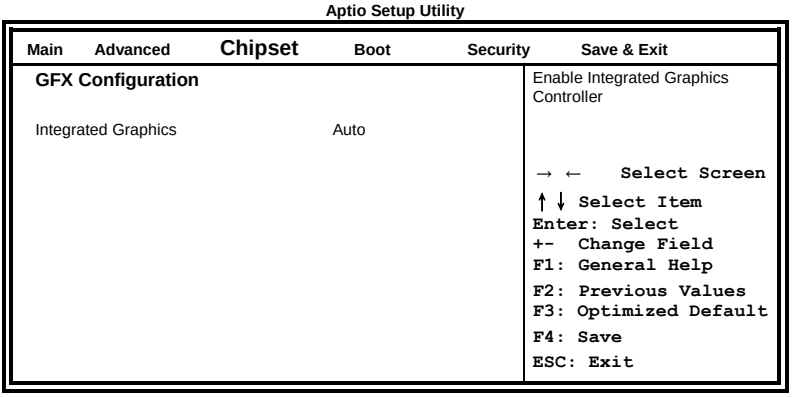
SATA IDE Combined Mode
Enabled or Disabled.

SB USB Configuration Options:

Main	Advanced	Chipset	Boot	Security	Save & Exit
XHCI Controller 0			Enabled		
XHCI Controller 1			Enabled		
DHCI HC(Bus 0 Dev 18 Fn 0)			Enabled		
EHCI HC(Bus 0 Dev 18 Fn 2)			Enabled		
DHCI HC(Bus 0 Dev 19 Fn 0)			Enabled		
EDHCI HC(Bus 0 Dev 19 Fn 0)			Enabled		
DHCI HC(Bus 0 Dev 20 Fn 5)			Enabled		
USB Port 0			Enabled		
USB Port			Enabled		
USB Port			Enabled		
USB Port			Enabled		
USB Port			Enabled		
			Enabled		→ ← Select Screen
USB Port			Enabled		↑ ↓ Select Item
USB Port			Enabled		Enter: Select
USB Port			Enabled		+- Change Field
USB Port			Enabled		F1: General Help
USB Port			Enabled		F2: Previous Values
					F3: Optimized Default
XHCI0 Port 0			Enabled		F4: Save
XHCI0 Port 1			Enabled		ESC: Exit
XHCI1 Port 0			Enabled		
XHCI1 Port 1			Enabled		

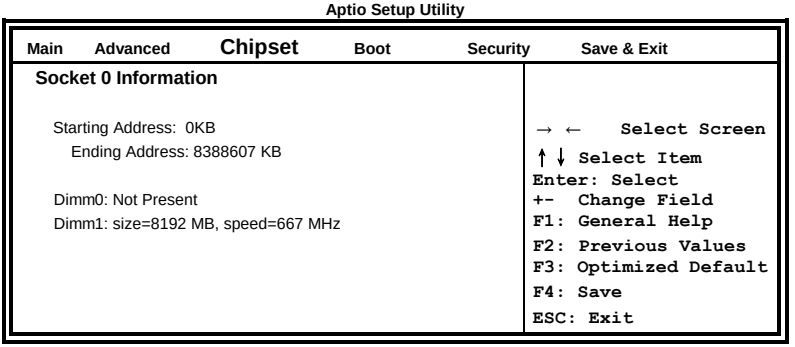
Aptio Setup Utility

Main	Advanced	Chipset	Boot	Security	Save & Exit
North Bridge Configuration					
► GFX Configuration					→ ← Select Screen
Memory Iniformation					↑ ↓ Select Item
Total memory: 8176 MB (DDR3)					Enter: Select
► Socket 0 Information					+- Change Field
					F1: General Help
					F2: Previous Values
					F3: Optimized Default
					F4: Save
					ESC: Exit



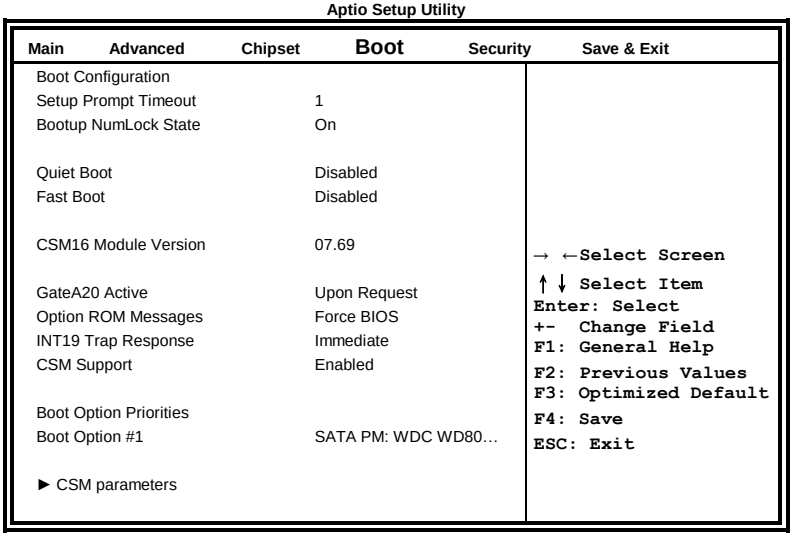
Integrated Graphics

Options are Auto Disabled and Force



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.

GateA20 Active

UPON REQUEST – GA20 can be disabled using BIOS services.
ALWAYS – do not allow disabling GA20; this option is useful when any RT code is executed above 1MB.

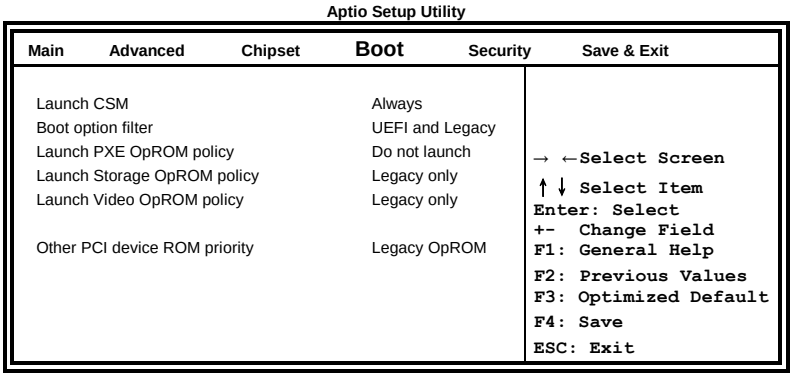
Option ROM Messages

Set display mode for Option ROM. Options are Force BIOS and Keep Current.

INT19 Trap Response
Enable: Allows Option ROMs to trap Int 19.

Boot Option Priorities
Sets the system boot order.

CSM parameters
OpROM execution, boot options, filter, etc.



Launch CSM
This option controls if CSM will be launched.

Boot option filter
This option controls what devices system can boot to.

Launch PXE OpROM policy
Controls the execution of UEFI and Legacy PXE OpROM.

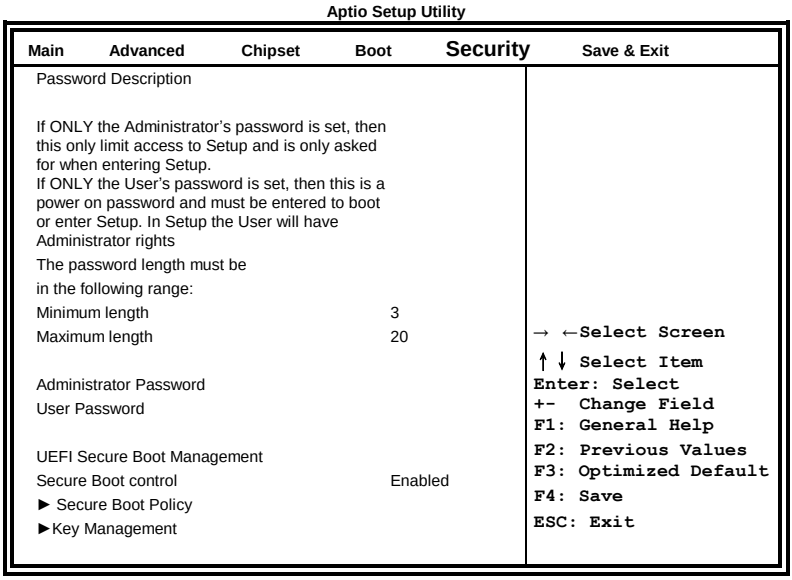
Launch Storatge OpROM policy
Controls the execution of UEFI and Legacy Storage OpROM.

Launch Video OpROM policy
Controls the execution of UEFI and Legacy Video OpROM.

Other PCI device ROM priority
For PCI devices other than Network, Mass storage or Video defines which OpROM to launch.

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.



Administrator Password

Set Setup Administrator Password.

User Password

Set User Password.

Secure Boot control

Secure Boot flow control.
Secure Boot is possible only if System runs in User Mode.

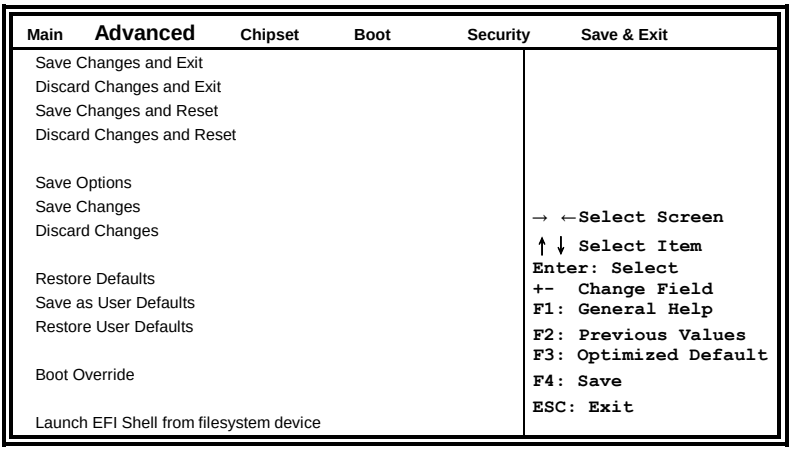
Secure Boot Policy

Select Secure Boot mode extended options: Internal FV, Option ROM, Removable Media, Fixed Media.

Administrator Password

Set Setup Administrator Password.

Save & Exit Settings



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.

Launch EFI Shell from filesystem device
Attempts to launch EFI Shell application (Shellx64.efi) from one of the available filesystem devices.

Drivers Installation

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

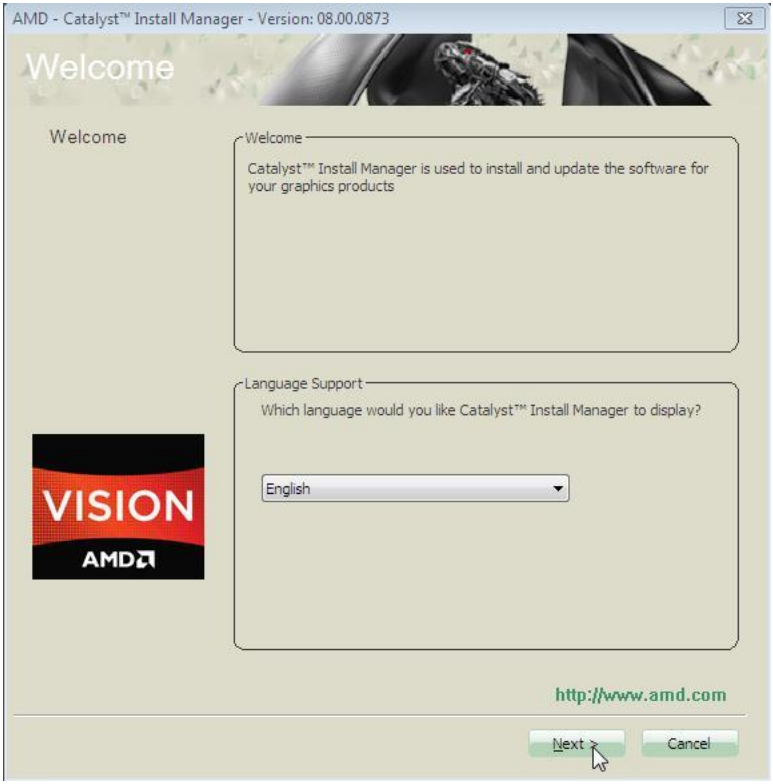
IMPORTANT NOTE:
After installing your Windows operating system, you must install first the Intel Chipset Software Installation Utility before proceeding with the drivers installation.

VGA Drivers Installation

1. Insert the drivers DVD that comes with the board. Click ***AMD***, then ***AMD eKabini Chipset Drivers***.

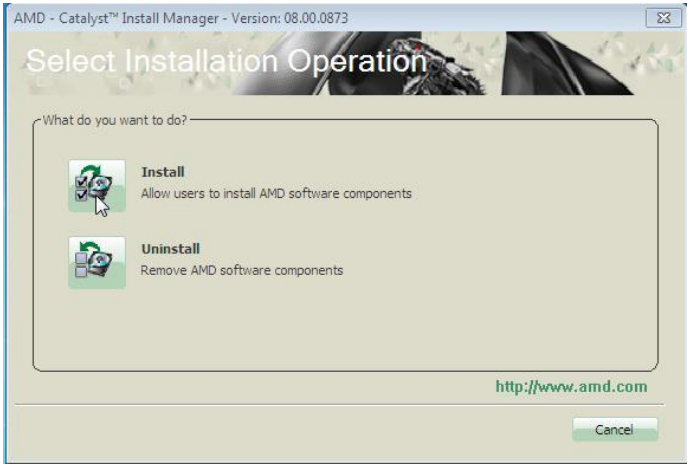
2. Click ***AMD eKabini Series Graphics Drivers***.

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

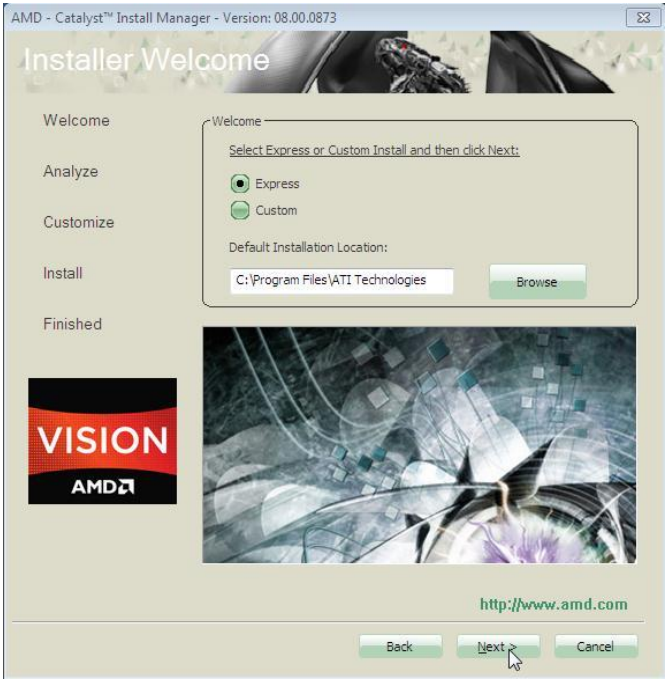


4. Select the language you would like to be displayed and click ***Next***.

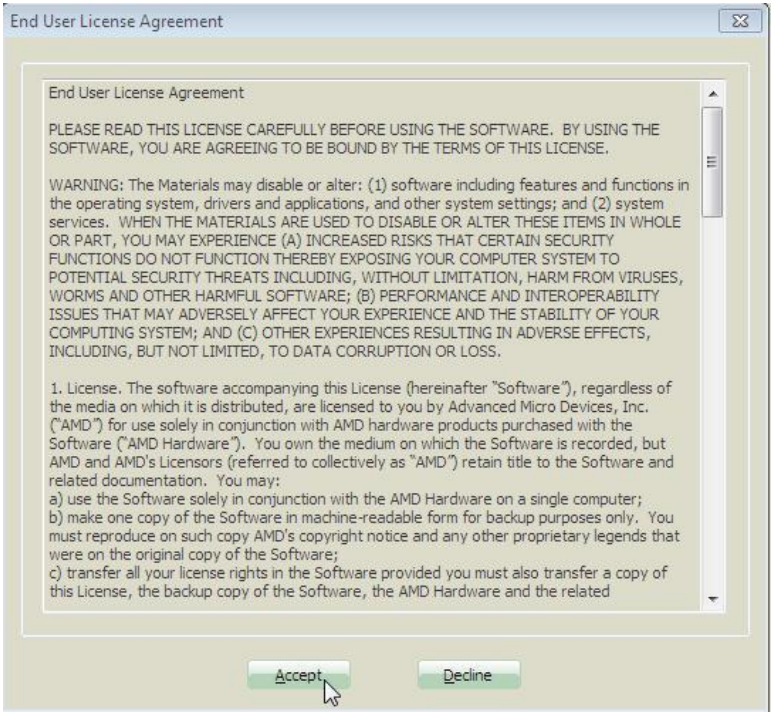
5. Click ***Next*** to continue the installation process.



6. Select **Express** and the **installation location** and click ***Next***.



7. Click ***Accept*** to accept the End User License Agreement.

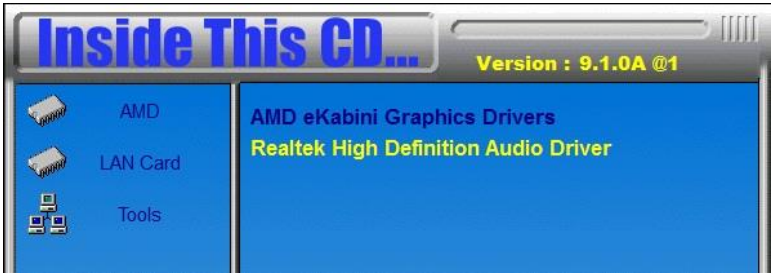


8. To reboot the system, click ***Yes***.

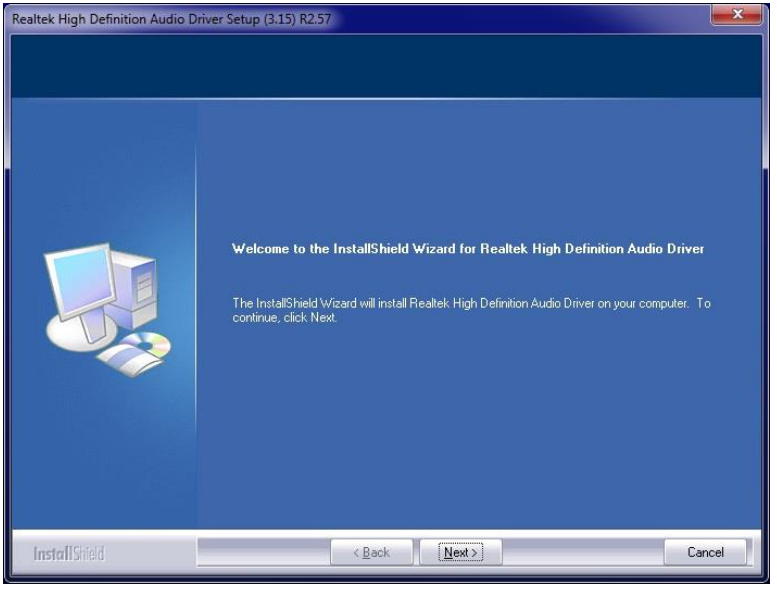
.

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.

2. Click ***LAN Card*** and then ***Realtek RTL8111G LAN Controller 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***.

Appendix

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
0000h-03AFh	PCI bus
0000h-03AFh	Direct memory access controller
0010h-001Fh	Motherboard resources
0020h-0021h	Programmable interrupt controller
0022h-003Fh	Motherboard resources
0040h-0043h	System timer
0044h-005Fh	Motherboard resources
0061h-0061h	System speaker
0063h-0063h	Motherboard resources
0065h-0065h	Motherboard resources
0070h-0071h	System CMOS/real time clock
0072h-007Fh	Motherboard resources
0081h-0083h	Direct memory access controller
0084h-0086h	Motherboard resources
0084h-0087h	Direct memory access controller
00A0h-00A1h	Programmable interrupt controller
00A2h-00BFh	Motherboard resources
00A2h-00BFh	Direct memory access controller
00B1h-00B1h	Motherboard resources
00F0h-00FFh	Numeric data processor
0170h-0177h	ATA Channel 1
01F0h-01F7h	ATA Channel 0
02E8H-02EFh	Communications Port (COM4)
02F8H-02FFh	Communications Port (COM2)
03E8H-03EFh	Communications Port (COM3)
03F8H-03FFh	Communications Port (COM1)

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 101/102-Key
IRQ 3	Communications Port (COM2)
IRQ 4	Communications Port (COM1)
IRQ 6	Communications Port (COM3)
IRQ 6	Communications Port (COM4)
IRQ 8	System CMOS/real time clock
IRQ 12	PS/2 Compatible Mouse
IRQ 13	Numeric data processor
IRQ 16	High Definition Audio Controller
IRQ 16	PCI standard PCI-to-PCI bridge
IRQ 17	Standard Enhanced PCI to USB Host Controller
IRQ 17	Standard Enhanced PCI to USB Host Controller
IRQ 18	High Definition Audio Controller
IRQ 18	Standard Open HCD USB Host Controller
IRQ 18	Standard Open HCD USB Host Controller
IRQ 18	Standard Open HCD USB Host Controller
IRQ 18	Standard Open HCD USB Host Controller
IRQ 19	PCI standard PCI-to-PCI bridge
IRQ 19	AMD SATA Controller (IDE Mode)

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:

```
//---------------------------------------------------------------------------
//
// 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 "F81846.H"
//---------------------------------------------------------------------------
int main (int argc, char *argv[]);
void EnableWDT(int);
void DisableWDT(void);
//---------------------------------------------------------------------------
int main (int argc, char *argv[])
{
        unsigned char bBuf;
        unsigned char bTime;
        char **endptr;

        char SIO;

        printf("Fintek 81866 watch dog program\n");

        SIO = Init_F81846();
        if (SIO == 0)
        {
                printf("Can not detect Fintek 81866, program abort.\n");
                return(1);
        }//if (SIO == 0)

        if (argc != 2)
        {
                printf(" Parameter incorrect!!\n");
                return (1);
        }

        bTime = strtol (argv[1], endptr, 10);
        printf("System will reset after %d seconds\n", bTime);

        if (bTime)
        {       EnableWDT(bTime);  }
        else
        {       DisableWDT();          }

        return 0;
```

```
}
//---------------------------------------------------------------------------
void EnableWDT(int interval)
{
	unsigned char bBuf;

	bBuf = Get_F81846_Reg(0x2B);
	bBuf &= (~0x20);
	Set_F81846_Reg(0x2B, bBuf);			//Enable WDTO

	Set_F81846_LD(0x07);				//switch to logic device 7
	Set_F81846_Reg(0x30, 0x01);			//enable timer

	bBuf = Get_F81846_Reg(0xF5);
	bBuf &= (~0x0F);
	bBuf |= 0x52;
	Set_F81846_Reg(0xF5, bBuf);			//count mode is second

	Set_F81846_Reg(0xF6, interval);		//set timer

	bBuf = Get_F81846_Reg(0xFA);
	bBuf |= 0x01;
	Set_F81846_Reg(0xFA, bBuf);			//enable WDTO output

	bBuf = Get_F81846_Reg(0xF5);
	bBuf |= 0x20;
	Set_F81846_Reg(0xF5, bBuf);			//start counting
}
//---------------------------------------------------------------------------
void DisableWDT(void)
{
	unsigned char bBuf;

	Set_F81846_LD(0x07);				//switch to logic device 7

	bBuf = Get_F81846_Reg(0xFA);
	bBuf &= ~0x01;
	Set_F81846_Reg(0xFA, bBuf);			//disable WDTO output

	bBuf = Get_F81846_Reg(0xF5);
	bBuf &= ~0x20;
	bBuf |= 0x40;
	Set_F81846_Reg(0xF5, bBuf);			//disable WDT
}
//---------------------------------------------------------------------------
```

```
//---------------------------------------------------------------------------
//
// 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 "F81846.H"
#include <dos.h>
//---------------------------------------------------------------------------
unsigned int F81846_BASE;
void Unlock_F81846 (void);
void Lock_F81846 (void);
//---------------------------------------------------------------------------
unsigned int Init_F81846(void)
{
	unsigned int result;
	unsigned char ucDid;

	F81846_BASE = 0x4E;
	result = F81846_BASE;

	ucDid = Get_F81846_Reg(0x20);
	if (ucDid == 0x07)                                  //Fintek 81866
	{	goto Init_Finish;	}

	F81846_BASE = 0x2E;
	result = F81846_BASE;

	ucDid = Get_F81846_Reg(0x20);
	if (ucDid == 0x07)                                  //Fintek 81866
	{	goto Init_Finish;	}

	F81846_BASE = 0x00;
	result = F81846_BASE;

Init_Finish:
	return (result);
}
//---------------------------------------------------------------------------
void Unlock_F81846 (void)
{
	outportb(F81846_INDEX_PORT, F81846_UNLOCK);
	outportb(F81846_INDEX_PORT, F81846_UNLOCK);
}
//---------------------------------------------------------------------------
void Lock_F81846 (void)
{
	outportb(F81846_INDEX_PORT, F81846_LOCK);
}
//---------------------------------------------------------------------------
void Set_F81846_LD( unsigned char LD)
{
	Unlock_F81846();
	outportb(F81846_INDEX_PORT, F81846_REG_LD);
	outportb(F81846_DATA_PORT, LD);
	Lock_F81846();
}
//---------------------------------------------------------------------------
void Set_F81846_Reg( unsigned char REG, unsigned char DATA)
{
	Unlock_F81846();
	outportb(F81846_INDEX_PORT, REG);
	outportb(F81846_DATA_PORT, DATA);
	Lock_F81846();
}
//---------------------------------------------------------------------------
```

```
unsigned char Get_F81846_Reg(unsigned char REG)
{
        unsigned char Result;
        Unlock_F81846();
        outportb(F81846_INDEX_PORT, REG);
        Result = inportb(F81846_DATA_PORT);
        Lock_F81846();
        return Result;
}
//---------------------------------------------------------------------------
```

```
//---------------------------------------------------------------------------
//
// 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 __F81846_H
#define __F81846_H                     1
//---------------------------------------------------------------------------
#define F81846_INDEX_PORT       (F81846_BASE)
#define F81846_DATA_PORT        (F81846_BASE+1)
//---------------------------------------------------------------------------
#define F81846_REG_LD           0x07
//---------------------------------------------------------------------------
#define F81846_UNLOCK           0x87
#define F81846_LOCK                     0xAA
//---------------------------------------------------------------------------
unsigned int Init_F81846(void);
void Set_F81846_LD( unsigned char);
void Set_F81846_Reg( unsigned char, unsigned char);
unsigned char Get_F81846_Reg( unsigned char);
//---------------------------------------------------------------------------
#endif  //__F81846_H
```

This page is intentionally left blank.