MI802

Intel® Cedar Trail + NM10
Mini-ITX Motherboard

USER'S MANUAL

Version 1.0A

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

MI802 is a Mini ITX board (170mm x 170mm) that comes with the Intel Atom D2550 processor that runs 1.86GHz and has a 1MB L2 cache. The board is based on the Intel® CG82NM10 PCH chipset and supports two SODIMM for a maximum memory support of 4GB.

MI802 features the integrated GMA3650 Intel graphics controller and supports both CRT and DVD-D video display interface, as well asa 24-bit dual channel interface with the help of Chrontel CH7511B device.

Onboard functional connections are available for two SATA ports, six COM ports, eight USB 2.0 ports, audio and a PS/2 keyboard/mouse stack connector. Additional Expansion comes in one slim PCI slot, two Mini PCI-e(x1) slots. Power connections is with a +12~24V DC in (ATX 4-pin).

Remarks: Currently, the board comes in two versions:
MI802 – with Cedar Trail-D / Atom D2550 1.86G TDP=10W
DDR3-1066MHz SO-DIMM x 2 (w/o ECC), Max. 4GB, Single channel

MI802N – with Cedar Trail-M / Atom N2600 1.6G TDP=3.5W
DDR3-800MHz SO-DIMM x 1 (w/o ECC), Max. 2GB, Single channel
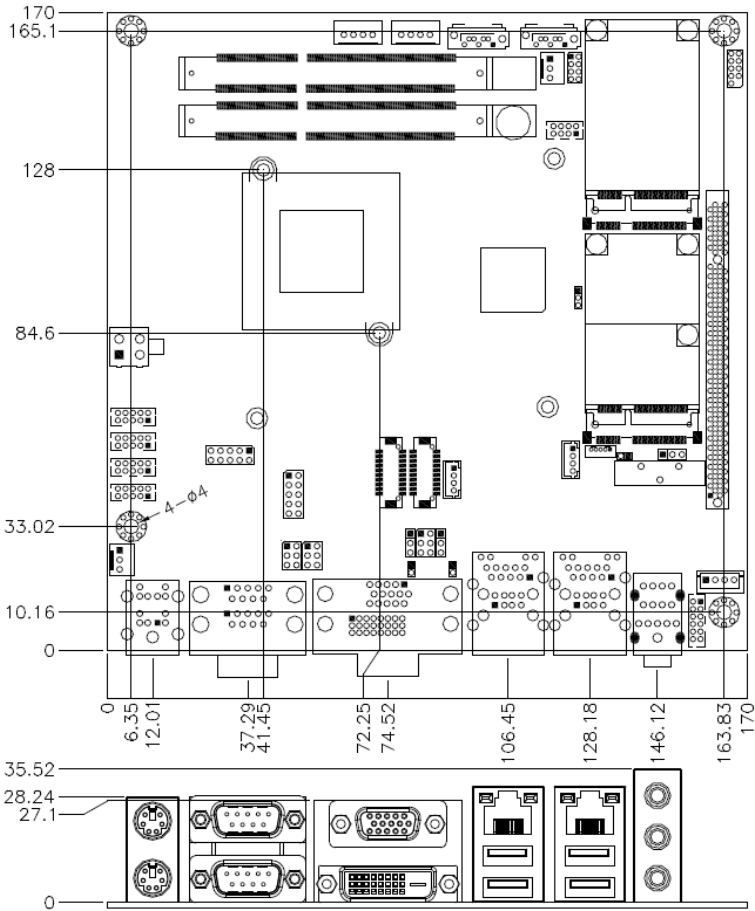
Checklist

Your MI802 package should include the items listed below.

- The MI802 Mini-ITX motherboard
- This User's Manual
- 1 CD containing chipset drivers and flash memory utility
- Serial ATA cable
- I/O shield

MI802 Specifications

Form Factor	Mini-ITX
CPU Type	Intel® Atom™ D2550/N2600 processor (32nm Technology) 22mm x 22mm, Micro-FCBGA559 [TDP=10W/3/5W]
CPU Speed	1.86GHz / 1MB L2 cache (MI802) 1.6GHz / 1MB L2 cache (MI802N)
BIOS	AMI BIOS, support ACPI Function
Chipset	Intel® CG82NM10 PCH: 17mm x 17mm, 360-pin BGA (2.1W)
Memory	DDR3-1066MHz SO-DIMM x 2 (w/o ECC), Max. 4GB , Single channel (MI802) DDR3-800MHz SO-DIMM x 1 (w/o ECC), Max. 2GB , Single channel (MI802N)
VGA	Intel® Integrated Graphics Controller (GMA3650) Supports DirectX 9 Graphic, OpenGL 3.0 Interface: CRT and DVI-D (thru level shifter ASM1442)
LVDS	Chrontel CH7511B for 24-bit dual channel LVDS interface
LAN	Realtek 8111E-VL GbE x 2
USB	NM10 built-in USB 2.0 host controller, support 8 ports - 4 ports in the rear panel - 3 ports for onboard pin header (3rd port signal shared with Mini PCIe) - 2 ports via Mini PCIe slot
Serial ATA Ports	NM10 built-in SATA controller, supports 2 ports
Audio	NM10 built-in HD audio controller w/ Realtek ALC269QHD Codec
LPC I/O	Fintek F81866AD-I: - COM1 (RS-232/422/485), COM2 (RS-232), COM3 (RS-232), COM4 (RS-232), COM5 (RS-232), COM6 (RS-232) - Hardware monitor (2 thermal inputs, 4 voltage monitor inputs, VID0~4 & 2 x Fan Header) - COM1/2 with pin-9 with power for 2 ports (500 mA for each port)
Digital IO	4 in & 4 out
Keyboard/Mouse Connector	Yes
Expansion Slots	Slim PCI slot x 1 Mini PCI-e(x1) x 2, Mounting holes for full-sized (x2) [Reserved mounting holes for Half-sized also x1]
Edge Connector	PS/2 KB+MS stack connector x 1 DB15 + DVI-D stack connector x 1 for CRT + DVI Dual DB9 stack connector x1 for COM#1 + COM#2 Dual USB + RJ45 x2 for USB + LAN HD Audio Jack 3x1 connector x 1 (Line-out, Line-in, MIC)
Onboard Header/ Connector	2x4 pin header x 1 for 2 USB ports 1x5 pin header x 1 for USB 4 [Shared with Mini PCIe USB signal] 2x6 pin header x 1 for front audio DF11 type 10 pin box header x 4 for COM # 3~6 2x5 pin header x 1 for Digital I/O DF13 box header x 2 for LVDS 4 pin box header x1 for LCD backlight control SATA connector x 2 4-pin HDD power connector x 2 for SATA device 4-pin ATX power connector for DC-in power 4-pin header for speaker out (from ALC269 internal amplifier)
Watchdog Timer	Yes (256 segments, 0, 1, 2…255 sec/min)
Power Connector	+12~24V DC-in (ATX 4-pins)
Others	1. iSMART function [Auto-scheduler & Power fail resume] 2. EuP/ErP thru Super I/O [WOL from 1ST GbE]
Board Size	170mm x 170mm

Board Dimensions

Installations

This section provides information on how to use the jumpers and connectors on the MI802 in order to set up a workable system. The topics covered are:

Installing the Memory

The MI802 board supports two DDR3-1066 memory.
The MI802N board supports one DDR3-800 memory.

Remarks:
D2550 supports SO-DIMM x 2 (w/o ECC), Max. 4GB, Single channel
N2600 supports SO-DIMM x 1 (w/o ECC), Max. 2GB, Single channel

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.
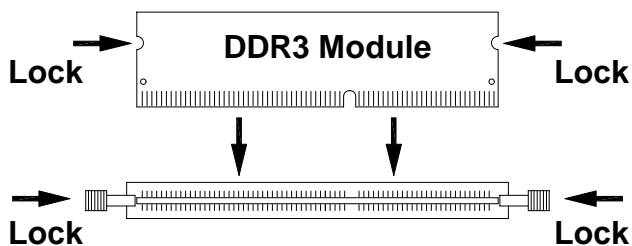
Setting the Jumpers

Jumpers are used on MI802 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 MI802 and their respective functions.
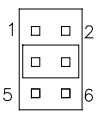
Jumper Locations on MI802

JP1: COM1 RS232 RI/+5V/+12V Power Setting

JP1	Setting	Function
	Pin 1-2 Short/Closed	+12V
	Pin 3-4 Short/Closed	RI*
	Pin 5-6 Short/Closed	+5V

JP2: COM2 RS232 RI/+5V/+12V Power Setting

JP2	Setting	Function
	Pin 1-2 Short/Closed	+12V
	Pin 3-4 Short/Closed	RI*
	Pin 5-6 Short/Closed	+5V

JP3: LCD Panel Power Selection

JP3	LCD Panel Power
	+3.3V*
	+5V

JP4: LCD Backlight Power Selection

JP4	Setting	Backlight Voltage
	Pin 1-2 Short/Closed	+3.3V
	Pin 3-4 Short/Closed	+5V
	Pin 5-6 Short/Closed	+12V*
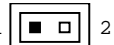
JP5: LCD BackLight Control Selection

JP5	Setting
1 2 3	Super I/O*
1 2 3	Reserved
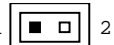
JP7: Clear CMOS Setting

JP7	Setting
1 2 3	Normal*
1 2 3	Clear CMOS

JP8: LCD BackLight Control Output Level

JP8	Setting
1 ■ □ 2	+3.3V*
1 ■ □ 2	+5V

JP9: LCD Boot ROM Protect (factory use only)

JP9	Setting
1 ■ □ 2	Write
1 ■ □ 2	Normal*

*Default

SW1: Panel Type Selection

8 7 6 5

ON
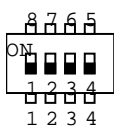
1 2 3 4

1 2 3 4

Default: Pin 1,2,3,4 OFF(1111)

SW1-4	SW1-3	SW1-2	SW1-1	Panel Type
ON	ON	ON	ON	800*600 18bit 1ch
ON	ON	ON	OFF	1024*768 18bit 1ch
ON	ON	OFF	ON	1024*768 24bit 1ch
ON	ON	OFF	OFF	1280*768 18bit 1ch
ON	OFF	ON	ON	1280*800 18bit 1ch
ON	OFF	ON	OFF	1280*960 18bit 1ch
ON	OFF	OFF	ON	1280*1024 24bit 2ch
ON	OFF	OFF	OFF	1366*768 18bit 1ch
OFF	ON	ON	ON	1366*768 24bit 1ch
OFF	ON	ON	OFF	1440*900 24bit 2ch
OFF	ON	OFF	ON	1440*1050 24bit 2ch
OFF	ON	OFF	OFF	1600*900 24bit 2ch
OFF	OFF	ON	ON	1680*1050 24bit 2ch
OFF	OFF	ON	OFF	1600*1200 24bit 2ch
OFF	OFF	OFF	ON	1920*1080 24bit 2ch
OFF	OFF	OFF	OFF	1920*1200 24bit 2ch

Connectors on MI802

The connectors on MI802 allows you to connect external devices such as keyboard, floppy disk drives, hard disk drives, printers, etc. The following table lists the connectors on MI802 and their respective functions.

Connector Locations on MI802

CN6: PS/2 Keyboard and PS/2 Mouse Connectors

 PS/2 Mouse
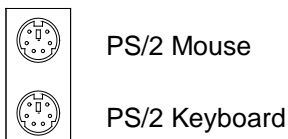
 PS/2 Keyboard

Signal Name	Keyboard	Mouse	Signal Name
Keyboard data	1	7	Mouse data
N.C.	2	8	N.C.
GND	3	9	GND
+5V	4	10	+5V
Keyboard clock	5	11	Mouse clock
N.C.	6	12	N.C.

CN7: COM1/RS232/422/485, COM2/RS232 Serial Port
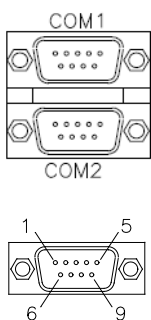
Pin #	Signal Name		
	RS-232	R2-422	RS-485
1	DCD	TX-	DATA-
2	RX	TX+	DATA+
3	TX	RX+	NC
4	DTR	RX-	NC
5	Ground	Ground	Ground
6	DSR	NC	NC
7	RTS	NC	NC
8	CTS	NC	NC
9*	RI	NC	NC

* Pin 9 supports RI/+5V/+12V function.

CN9: VGA and DVI-D Connector
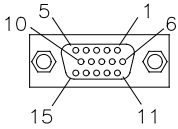
Signal Name	Pin #	Pin #	Signal Name
Red	1	2	Green
Blue	3	4	N.C.
GND	5	6	GND
GND	7	8	GND
+5V	9	10	GND
N.C.	11	12	DDC DATA
HSYNC	13	14	VSYNC
DDC CLOCK	15		

VGA
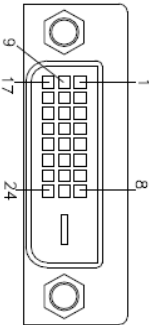
Signal Name	Pin #	Pin #	Signal Name
DATA 2-	1	16	HOT PLUG
DATA 2+	2	17	DATA 0-
Shield 2	3	18	DATA 0+
N.C.	4	19	SHIELD 0
N.C.	5	20	N.C.
DDC CLOCK	6	21	N.C.
DDC DATA	7	22	SHIELD CLK
N.C	8	23	CLOCK-
DATA 1-	9	24	CLOCK+
DATA 1+	10	C1	N.C.
SHIELD 1	11	C2	N.C.
N.C.	12	C3	N.C.
N.C.	13	C4	N.C.
+5V	14	C5	N.C.
GROUND	15	C6	N.C.

DVI-D

CN15: GbE 1st RJ-45 and USB0/1 Ports

CN19: GbE 2nd RJ-45 and USB2/3 Ports

CN22: Line-in, Line-out & Microphone Connector

L-IN
L-OUT
MIC
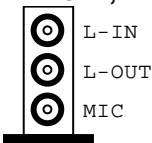
CN1, CN2, CN3, CN4: COM3/4/5/6 RS232 Serial Port

2X5_2.0mm_Straight_Male_DF11 (Haoguo DF11-10S-PA66H, Mating connector: Hirose DF11-10DS-2C or compatible)
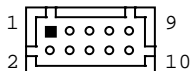
Signal Name	Pin #	Pin #	Signal Name
DCD, Data carrier detect	1	2	RXD, Receive data
TXD, Transmit data	3	4	DTR, Data terminal ready
GND, ground	5	6	DSR, Data set ready
RTS, Request to send	7	8	CTS, Clear to send
RI, Ring indicator	9	10	Not Used

CN5: ATX 4P +12V~24V Connector
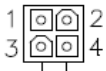
Signal Name	Pin #	Pin #	Signal Name
GND	1	2	GND
+12~24V	3	4	+12~24V

CN8, CN11: HDD Power Connector

1X4_2.5mm_Straight_Male_Wafer (Haoguo W7-03H104142S1WT)

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

CN10, CN12: LVDS Connectors (1st channel, 2nd channel)
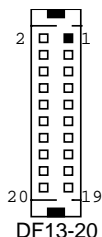
Signal Name	Pin #	Pin #	Signal Name
TX0-	2	1	TX0+
Ground	4	3	Ground
TX1-	6	5	TX1+
+5V/3.3V	8	7	Ground
TX3-	10	9	TX3+
TX2-	12	11	TX2+
Ground	14	13	Ground
TXC-	16	15	TXC+
+5V/3.3V	18	17	ENABKL
NC	20	19	NC

DF13-20

LVDS Connectors: 2X10_1.25mm_Straight_Male_DF13 (Hirose DF13-20DP-1.25V)
The LVDS (24bit) connectors on board consist of the first channel (LVDS1) and second channel (LVDS2).

CN13: LCD Backlight Connector

1X4_2.0mm_Straight_M(JST B4B-PH-K-S)
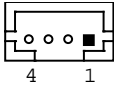
Pin #	Signal Name
1	+3.3V/5V/12V*
2	Backlight Enable
3	Brightness Control
4	Ground

CN14,CN16: SATA Connectors
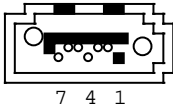
Pin #	Signal Name
1	Ground
2	TX+
3	TX-
4	Ground
5	RX-
6	RX+
7	Ground

CN17: USB6/7 Ports Header

2X4_2.0mm_Straight_Male_DF11 (Haoguo DF11-8S-PA66H, Mating connector: Hirose DF11-8DS-2C or compatible)
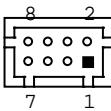
Signal Name	Pin #	Pin #	Signal Name
+5V	1	2	Ground
Data-	3	4	Data+
Data+	5	6	Data-
Ground	7	8	+5V

CN18: iSMART JTAG Pin Header (factory use only)

CN20, CN21: Mini PCIE Connector
Remarks: CN20 USB signal is shared with J5.

CN23: Audio Connector (DF11 Connector)

2X6_2.0mm_Straight_Male_DF11 (Haoguo DF11-12S-PA66H, Mating connector: Hirose DF11-12DS-2C or compatible)
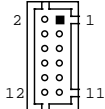
Signal Name	Pin #	Pin #	Signal Name
LINEOUT_R	2	1	LINEOUT_L
Ground	4	3	JD_LINEOUT
LINEIN_R	6	5	LINEIN_L
Ground	8	7	JD_LINEIN
MIC_R	10	9	MIC_L
Ground	12	11	JD_MIC1

CN24: Speaker Connector (JST Connector)

1X4_2.5mm_Straight_Male_Wafer (E-CALL 0110-071-040, Mating connector: JST XHP-4 or compatible)
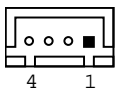
Pin #	Signal Name
1	Speaker-L+
2	Speaker-L-
3	Speaker-R-
4	Speaker-R+

CPU_FAN1: CPU Fan Power Connector

Pin #	Signal Name
1	Ground
2	+12V
3	Rotation detection

DIMM1, DIMM2: DDR3 SO-DIMM

J1: LPC Debug Connector (factory use only)

J2: Digital I/O
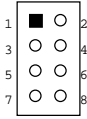
2X5_2.54mm_Straight_Male_Pin Header (E-CALL 0126-01-203-100)
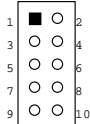
Signal Name	Pin #	Pin #	Signal Name
GND	1	2	+5V
OUT3	3	4	OUT1
OUT2	5	6	OUT0
IN3	7	8	IN1
IN2	9	10	IN0

J3: System Function Connector

2X4_2.0mm_Straight_Male_Pin Header (E-CALL 0196-01-200-080)
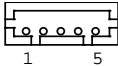
Signal Name	Pin #	Pin #	Signal Name
Power BTN	1	2	Power BTN
Power LED+	3	4	Power LED-
HDD LED+	5	6	HDD LED-
Reset BTN	7	8	Reset BTN

J5: USB4 Port

Remarks: Signal is shared with Mini PCIe(CN20)

1X5_1.25mm_Straight_Male_Wafer (E-CALL 0110-26110050, Mating connector: Molex 51021-0500 or compatible)

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

J6: SPI Flash Connector (factory use only)

PCI1: PCI Slot (supports 2 Master)

SYS_FAN1: System Fan Power Connector

Pin #	Signal Name
1	Ground
2	+12V
3	Rotation detection

This page is intentionally left blank.

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 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> 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    Advanced      Chipset      Boot       Security     Save & Exit

 BIOS Information
 System Date                    Sat 07/21/2012
 System Time                    13:03:22




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

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.

Aptio Setup Utility

Main	**Advanced**	Chipset	Boot	Security	Save & Exit

Legacy OpROM Support
Launch PXE OpROM   Disabled
Lannch Storage OpROM   Disabled

► PCI Subsystem Settings
► ACPI Settings
► Wake up event setting
► CPU Configuration
► Auto Power On Schedule
► IDE Configuration
► USB Configuration
► F81866 Super IO Configuration
► FB81866 H/W Monitor
► PPM Configuration

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

Launch PXE OpROM

Enable or Disable Boot Option for Legacy Network Devices.

PCI Subsystem Settings

Aptio Setup Utility

Main	**Advanced**	Chipset	Boot	Security	Save & Exit
PCI Bus Driver Version			V 2.05.01		
PCI ROM Priority			Legacy ROM		
PCI Common Settings					
PCI Latency Timer			32 PCI Bus Clocks		
VGA Palette Snoop			Disabled		
PERR# Generation			Disabled		
SERR# Generation			Disabled		

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

PCI ROM Priority
In case of multiple Option ROMs (Legacy and EFI Compatible), specifies what PCI Option ROM to launch.

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#.

ACPI Settings

Main	Advanced	Chipset	Boot	Security	Save & Exit
ACPI Settings					→ ←Select Screen
					↑↓ Select Item
Enable ACPI Auto Configuration		Disabled			Enter: Select
					+- Change Opt
Enable Hibernation		Enabled			F1: General Help
					F2: Previous Values
					F3: Optimized Default
					F4: Save & EXIT
					ESC: Exit

Enabled 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.

Wake Up Event Setting

Aptio Setup Utility

Main	**Advanced**	Chipset	Boot	Security	Save & Exit

Wake on Ring		Disabled	→ ←Select Screen
Wake on PCIE PME		Disabled	↑↓ Select Item
			Enter: Select
			+- Change Opt
			F1: General Help
			F2: Previous Values
			F3: Optimized Default
			F4: Save & EXIT
			ESC: Exit

Wake on Ring
The options are Disabled and Enabled.

Wake on PCIE PME
The options are Disabled and Enabled.

CPU Configuration

This section shows the CPU configuration parameters.

Aptio Setup Utility BIOS screen showing CPU Configuration under the Advanced tab

Main	**Advanced**	Chipset	Boot	Security	Save & Exit

```
CPU Configuration

Processor Type              Intel(R) Atom(TM) CPU
EMT64                       Not Supported
Processor Speed             1865 MHz
System Bus Speed            533 MHz
Ratio Status                14
Actual Ratio                14
System Bus Speed            533 MHz
Processor Stepping          30661
Microcode Revision          265
L1 Cache RAM                2x56 k                → ←Select Screen
L2 Cache RAM                2x512 k               ↑↓ Select Item
Processor Core              Dual             Enter: Select
Hyper-Threading             Supported        +-  Change Opt
                                             F1: General Help
Hyper-Threading             Enabled          F2: Previous Values
Execute Disable Bit         Enabled          F3: Optimized Default
Limit CPUID Maximum         Disabled         F4: Save & EXIT
                                             ESC: Exit
```

Hyper-threading

Enabled for Windows XP and Linux (OS optimized for Hyper-Threading Technology) and Disabled for other OS (OS not optimized for Hyper-Threading Technology). When Disabled, only one thread per enabled core is enabled.

Execute Disable Bit

XD can prevent certain classes of malicious buffer overflow attacks when combined with a supporting OS (Windows Server 2003 SP1, Windows XP SP2, SuSE Linux 9.2, Re33dHat Enterprise 3 Update 3.)

Limit CPUID Maximum

Disabled for Windows XP.

iSmart Controller

Aptio Setup Utility

Main	**Advanced**	Chipset	Boot	Security	Save & Exit
Auto Power On Schedule					→ ←Select Screen
Power-On after Power failure		Disable			↑↓ Select Item
					Enter: Select
Schedule Slot 1		None			+- Change Opt
Schedule Slot 2		None			F1: General Help
					F2: Previous Values
					F3: Optimized Default
					F4: Save & EXIT
					ESC: Exit

EuP/ErP Standby Power Control
Saving the power consumption on power off.

Power-On after Power Failure
This field sets the system power status whether *on or off* when power returns to the system from a power failure situation.

Schedule Slot
None / Power On / Power On/Off – Setup the hour/minute for system power on

IDE Configuration

Aptio Setup Utility

Main	**Advanced**	Chipset	Boot	Security	Save & Exit
SATA Port0		Not Present			→ ←Select Screen
SATA Port1		Not Present			↑↓ Select Item
					Enter: Select
SATA Controller(s)		Enabled			+- Change Opt
					F1: General Help
Configure SATA as		IDE			F2: Previous Values
					F3: Optimized Default
					F4: Save & EXIT
					ESC: Exit

SATA Controller(s)
Enable / Disable Serial ATA Controller.

Configure SATA as
(1) IDE Mode.
(2) AHCI Mode.

USB Configuration

Aptio Setup Utility

Main	**Advanced**	Chipset	Boot	Security	Save & Exit

```
USB Configuration

USB Devices:
    None

Legacy USB Support            Enabled
EHCI Hand-off                 Enabled

USB hardware delays and time-outs:
USB Transfer time-out         20 sec
Device reset time-out         20 sec
Device power-up delay         Auto
```

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

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

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.

Super IO Configuration

Aptio Setup Utility

Main	**Advanced**	Chipset	Boot	Security	Save & Exit

F81866 Super IO Configuration

F81866 Super IO Chip   F81866
► Serial Port 0 Configuration
► Serial Port 1 Configuration
► Serial Port 2 Configuration
► Serial Port 3 Configuration
► Serial Port 4 Configuration
► Serial Port 5 Configuration

Power Failure I   Always off
KB/MS Power on   None
F81866 ERP Support   All Enable

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

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.

H/W Monitor

Aptio Setup Utility

Main	Advanced	Chipset	Boot	Security	Save & Exit
PC Health Status					
CPU temperature		+52 C			
System temperature		+54 C			
Fan1 Speed		N/A			
Fan2 Speed		N/A			
Vcore		+1.208			→ ←Select Screen
VCC5V		+5.129V			↑↓ Select Item
VCC12V		+11.528 V			Enter: Select
+1.5V		+1.520 V			+- Change Opt
VSB5V		+5.040 V			F1: General Help
VCC3V		+3.392 V			F2: Previous Values
VSB3V		+3.376 V			F3: Optimized Default
CPU FAN1		Disabled			F4: Save & EXIT
SYS FAN1		Disabled			ESC: Exit

Temperatures/Voltages

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

CPU Fan1 Control

Options are: Disabled (default), 50 ℃, 60 ℃, 70 ℃, 80 ℃

SYS Fan2 Control

Options are: Disabled (default), 50 ℃, 60 ℃, 70 ℃, 80 ℃

PPM Configuration

Aptio Setup Utility

Main	Advanced	Chipset	Boot	Security	Save & Exit
PPM Configuration					
					→ ←Select Screen
EIST			Enabled		↑↓ Select Item
					Enter: Select
					+- Change Opt
					F1: General Help
					F2: Previous Values
					F3: Optimized Default
					F4: Save & EXIT
					ESC: Exit

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.

```
                        Aptio Setup Utility
┌──────────────────────────────────────────────────────────────────┐
│  Main    Advanced    Chipset    Boot     Security    Save & Exit  │
│                                                                    │
│  ► Host Bridge                                                     │
│  ► South Bridge                          → ←Select Screen          │
│                                          ↑ ↓ Select Item           │
│                                          Enter: Select             │
│                                          +-  Change Field          │
│                                          F1: General Help          │
│                                          F2: Previous Values       │
│                                          F3: Optimized Default     │
│                                          F4: Save                  │
│                                          ESC: Exit                 │
└──────────────────────────────────────────────────────────────────┘
```

Host Bridge
This item shows the Host Bridge Parameters.

South Bridge
This item shows the South Bridge Parameters.

Host Bridge
This section allows you to configure the Host Bridge Chipset.

```
                        Aptio Setup Utility
┌──────────────────────────────────────────────────────────────────┐
│  Main    Advanced    Chipset    Boot     Security    Save & Exit  │
│                                                                    │
│  ► Memory Frequency and Timing                                     │
│  ► Intel IGD Configuration               → ←   Select Screen       │
│                                          ↑ ↓ Select Item           │
│  ***********Memory Information***********  Enter: Select           │
│  Memory Frequency        1067 MHz(DDR3)  +-  Change Opt            │
│  Total Memory            2048 MB         F1: General Help          │
│  DIMM#0                  Not Present     F2: Previous Values       │
│  DIMM#1                  2048 MB         F3: Optimized Default     │
│                                          F4: Save & EXIT           │
│                                          ESC: Exit                 │
└──────────────────────────────────────────────────────────────────┘
```

Memory Frequency and Timing

Main	Advanced	Chipset	Boot	Security	Save & Exit

```
Memory Frequency and Timing

MRC Fast Boot              Enabled
Max TOLUD                  Dynamic          → ←Select Screen
                                            ↑↓ Select Item
                                            Enter: Select
                                            +-  Change Opt
                                            F1: General Help
                                            F2: Previous Values
                                            F3: Optimized Default
                                            F4: Save & EXIT
                                            ESC: Exit
```

MRC Fast Boot
The options are Disabled and Enabled.

Max TOLUD
The default setting is Dynamic.

Intel IGD Configuration

Aptio Setup Utility

Main	Advanced	Chipset	Boot	Security	Save & Exit

```
Intel IGD Configuration
IGFX-Boot Type             VBIOS Default

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

IGFX-Boot Type
Select the video Device which will be activated during POST.

South Bridge

This section allows you to configure the South Bridge Chipset.

Aptio Setup Utility

Main	Advanced	Chipset	Boot	Security	Save & Exit

```
► TPT Device
► PCI Express Root Port0
► PCI Express Root Port1
► PCI Express Root Port2
► PCI Express Root Port3
                                                      → ←Select Screen
DMI Link ASPM Control             Enabled             ↑ ↓ Select Item
PCI-Exp. High Priority Port       Disabled            Enter: Select
                                                      +-   Change Opt
High Precision Event Timer Configuration              F1: General Help
High Precision Timer              Enabled             F2: Previous Values
                                                      F3: Optimized Default
SLP_SP4 Assertion Width           1-2 Seconds         F4: Save & EXIT
                                                      ESC: Exit
```

DMI Clink ASPM Control

The control of Active State Power Management on both NB side and SB side of the DMI Link.

PCI-Exp. High Priority Port

The options are Disabled, Port1, Port2, Port3, and Port4.

High Precision Event Timer Configuration

Enable/or Disable the High Precision Event Timer.

SLP_S4 Assertion Stretch Enable

Select a minimum assertion width of the SLP_S4# signal.

TPT Device

Aptio Setup Utility

Main	Advanced	Chipset	Boot	Security	Save & Exit

Azalia Controller	HD Audio	→ ←**Select Screen**
		↑↓ **Select Item**
Select USB Mode	By Controllers	**Enter: Select**
UHCI #1 (port 0 and 1)	Enabled	**+- Change Opt**
UHCI #2 (port 2 and 3)	Enabled	**F1: General Help**
UHCI #3 (port 4 and 5)	Enabled	**F2: Previous Values**
UHCI #4 (port 6 and 7)	Enabled	**F3: Optimized Default**
USB 2.0(EHCI) Support	Enabled	**F4: Save & EXIT**
		ESC: Exit

PCI Express Root Port0

Aptio Setup Utility

Main	Advanced	Chipset	Boot	Security	Save & Exit

PCI Express Port 0	Enabled	→ ←**Select Screen**
Port 0 IOxAPIC	Disabled	↑↓ **Select Item**
Automatic ASPM	Manual	**Enter: Select**
ASPM L0s	Root Port Only	**+- Change Opt**
ASPM L1	Enabled	**F1: General Help**
		F2: Previous Values
		F3: Optimized Default
		F4: Save & EXIT
		ESC: Exit

PCI Express Root Port1

Aptio Setup Utility

Main	Advanced	Chipset	Boot	Security	Save & Exit

PCI Express Port 1	Auto	→ ←**Select Screen**
Port 0 IOxAPIC	Disabled	↑↓ **Select Item**
Automatic ASPM	Auto	**Enter: Select**
		+- Change Opt
		F1: General Help
		F2: Previous Values
		F3: Optimized Default
		F4: Save & EXIT
		ESC: Exit

PCI Express Root Port2

Main	Advanced	Chipset	Boot	Security	Save & Exit

PCI Express Port 2	Auto	→ ←Select Screen
Port 0 IOxAPIC	Disabled	↑ ↓ Select Item
Automatic ASPM	Auto	Enter: Select
		+- Change Opt
		F1: General Help
		F2: Previous Values
		F3: Optimized Default
		F4: Save & EXIT
		ESC: Exit

PCI Express Root Port3

Main	Advanced	Chipset	Boot	Security	Save & Exit

PCI Express Port 3	Enabled	→ ←Select Screen
Port 0 IOxAPIC	Disabled	↑ ↓ Select Item
Automatic ASPM	Auto	Enter: Select
		+- Change Opt
		F1: General Help
		F2: Previous Values
		F3: Optimized Default
		F4: Save & EXIT
		ESC: Exit

Boot Settings

Aptio Setup Utility					
Main	Advanced	Chipset	**Boot**	Security	Save & Exit

```
Boot Configuration
Setup Prompt Timeout              1
Bootup NumLock State             On

Quiet Boot                       Disabled
Fast Boot                        Disabled

CSM16 Module Version             07.68

GateA20 Active                   Upon Request
Option ROM Messages              Force BIOS
Interrupt 19 Canture             Enabled
CSM Support                      Enabled

Boot Option Priorities
```

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

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: Force BIOS and Keep Current.

Interrupt 19 Capture
Enable: Allows Option ROMs to trap Int 19.

CSM Support
Enables/Disables/Auto CSM Support.

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 limits 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

Administrator Password
User Password

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

Administrator Password
Set Setup Administrator Password.

User Password
Set User Password.

Save & Exit Settings

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

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) Cedarview 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 *Yes* to accept the software license agreement and proceed with the installation process.

5. On the Readme File Information screen, click *Next* 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) Cedarview Graphics Driver.*

2. When the Welcome screen appears, click *Next* to continue.
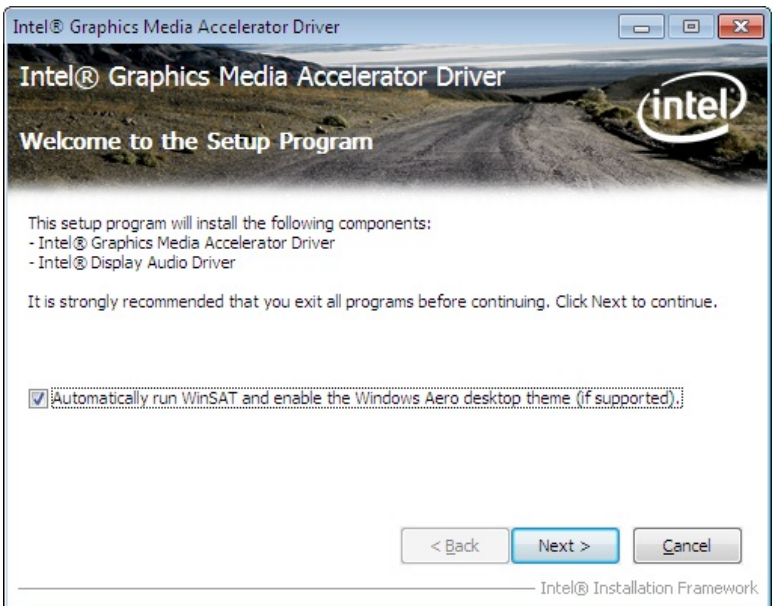
4. Click *Yes* to to agree with the license agreement and continue the installation.

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

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

7. 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.
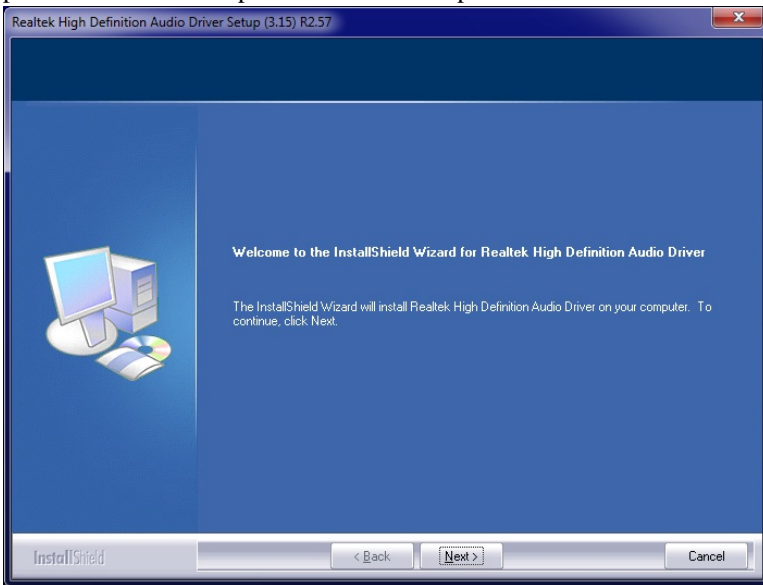
3. Restart the computer when prompted.

LAN Drivers Installation

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

2. Click *Realtek RTL8111E LAN 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*.

This page is intentionally left blank.

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 - 01Fh	DMA Controller #1
020h - 03Fh	Interrupt Controller #1
040h - 05Fh	Timer
060h - 06Fh	Keyboard Controller
070h - 07Fh	Real Time Clock, NMI
080h - 09Fh	DMA Page Register
0A0h - 0BFh	Interrupt Controller #2
0C0h - 0DFh	DMA Controller #2
0F0h	Clear Math Coprocessor Busy Signal
0F1h	Reset Math Coprocessor
1F0h - 1F7h	IDE Interface
2B0h - 2DFh	Graphics adapter Controller
2E8h - 2EFh	Serial Port #4(COM4)
2F8h - 2FFh	Serial Port #2(COM2)
360h - 36Fh	Network Ports
3B0h - 3BFh	Monochrome & Printer adapter
3C0h - 3CFh	EGA adapter
3D0h - 3DFh	CGA adapter
3E8h - 3EFh	Serial Port #3(COM3)
3F8h - 3FFh	Serial Port #1(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
IRQ0	System Timer Output
IRQ1	Keyboard
IRQ2	Interrupt Cascade
IRQ3	Serial Port #2
IRQ4	Serial Port #1
IRQ5	Reserved
IRQ6	Reserved
IRQ7	Reserved
IRQ8	Real Time Clock
IRQ9	Reserved
IRQ10	Serial Port #4
IRQ11	Serial Port #3
IRQ12	PS/2 Mouse
IRQ13	80287
IRQ14	Primary IDE

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:

```
//---------------------------------------------------------------------------
#include <dos.h>
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#include "F81866.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 81865 watch dog program\n");

        SIO = Init_F81866();
        if (SIO == 0)
        {
                printf("Can not detect Fintek 81865, 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_F81866_Reg(0x2B);
        bBuf &= (~0x20);
        Set_F81866_Reg(0x2B, bBuf);                        //Enable WDTO

        Set_F81866_LD(0x07);                               //switch to logic device 7
        Set_F81866_Reg(0x30, 0x01);                        //enable timer

        bBuf = Get_F81866_Reg(0xF5);
        bBuf &= (~0x0F);
        bBuf |= 0x52;
        Set_F81866_Reg(0xF5, bBuf);                        //count mode is second

        Set_F81866_Reg(0xF6, interval);            //set timer

        bBuf = Get_F81866_Reg(0xFA);
        bBuf |= 0x01;
        Set_F81866_Reg(0xFA, bBuf);                        //enable WDTO output

        bBuf = Get_F81866_Reg(0xF5);
        bBuf |= 0x20;
        Set_F81866_Reg(0xF5, bBuf);                        //start counting
}
//-------------------------------------------------------------------------
void DisableWDT(void)
{
        unsigned char bBuf;

        Set_F81866_LD(0x07);                               //switch to logic device 7

        bBuf = Get_F81866_Reg(0xFA);
        bBuf &= ~0x01;
        Set_F81866_Reg(0xFA, bBuf);                        //disable WDTO output

        bBuf = Get_F81866_Reg(0xF5);
        bBuf &= ~0x20;
        bBuf |= 0x40;
        Set_F81866_Reg(0xF5, bBuf);                        //disable WDT
}
//-------------------------------------------------------------------------
```

```
//-------------------------------------------------------------------------
#include "F81866.H"
#include <dos.h>
//-------------------------------------------------------------------------
unsigned int F81866_BASE;
void Unlock_F81866 (void);
void Lock_F81866 (void);
//-------------------------------------------------------------------------
unsigned int Init_F81866(void)
{
        unsigned int result;
        unsigned char ucDid;

        F81866_BASE = 0x4E;
        result = F81866_BASE;

        ucDid = Get_F81866_Reg(0x20);
        if (ucDid == 0x07)                              //Fintek 81865
        {       goto Init_Finish;       }

        F81866_BASE = 0x2E;
        result = F81866_BASE;

        ucDid = Get_F81866_Reg(0x20);
        if (ucDid == 0x07)                              //Fintek 81865
        {       goto Init_Finish;       }

        F81866_BASE = 0x00;
        result = F81866_BASE;

Init_Finish:
        return (result);
}
//-------------------------------------------------------------------------
void Unlock_F81866 (void)
{
        outportb(F81866_INDEX_PORT, F81866_UNLOCK);
        outportb(F81866_INDEX_PORT, F81866_UNLOCK);
}
//-------------------------------------------------------------------------
void Lock_F81866 (void)
{
        outportb(F81866_INDEX_PORT, F81866_LOCK);
}
//-------------------------------------------------------------------------
void Set_F81866_LD( unsigned char LD)
{
        Unlock_F81866();
        outportb(F81866_INDEX_PORT, F81866_REG_LD);
        outportb(F81866_DATA_PORT, LD);
        Lock_F81866();
}
//-------------------------------------------------------------------------
void Set_F81866_Reg( unsigned char REG, unsigned char DATA)
{
        Unlock_F81866();
        outportb(F81866_INDEX_PORT, REG);
        outportb(F81866_DATA_PORT, DATA);
        Lock_F81866();
}
//-------------------------------------------------------------------------
unsigned char Get_F81866_Reg(unsigned char REG)
{
        unsigned char Result;
        Unlock_F81866();
        outportb(F81866_INDEX_PORT, REG);
        Result = inportb(F81866_DATA_PORT);
        Lock_F81866();
        return Result;
}
//-------------------------------------------------------------------------
```

```
//----------------------------------------------------------------------------
#ifndef __F81866_H
#define __F81866_H                      1
//----------------------------------------------------------------------------
#define      F81866_INDEX_PORT          (F81866_BASE)
#define      F81866_DATA_PORT           (F81866_BASE+1)
//----------------------------------------------------------------------------
#define      F81866_REG_LD              0x07
//----------------------------------------------------------------------------
#define F81866_UNLOCK             0x87
#define      F81866_LOCK                          0xAA
//----------------------------------------------------------------------------
unsigned int Init_F81866(void);
void Set_F81866_LD( unsigned char);
void Set_F81866_Reg( unsigned char, unsigned char);
unsigned char Get_F81866_Reg( unsigned char);
//----------------------------------------------------------------------------
#endif  //__F81866_H
```

D. Digital I/O Sample Code

Files of the MAIN.CPP

```
//----------------------------------------------------------------------------
#include <dos.h>
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#include "F81866.H"
#define BIT0 0x01
#define BIT1 0x02
#define BIT2 0x04
#define BIT3 0x08
#define BIT4 0x10
#define BIT5 0x20
#define BIT6 0x40
#define BIT7 0x80
//----------------------------------------------------------------------------
int main (void);

void Dio3Initial(void);
void Dio3SetOutput(unsigned char);
unsigned char Dio3GetInput(void);
void Dio3SetDirection(unsigned char);
unsigned char Dio3GetDirection(void);
//----------------------------------------------------------------------------
int main (void)
{
      char SIO;
      unsigned char DIO;

      printf("Fintek 81866 digital I/O test program. Ver 2.0\n");

      SIO = Init_F81866();
      if (SIO == 0)
      {
            printf("Can not detect Fintek 81866, program abort.\n");
            return(1);
      }//if (SIO == 0)

      Dio3Initial();
/*
      //for GPIO30..37
      Dio3SetDirection(0xF0); //GP30..33 = input, GP34..37=output
      printf("Current DIO direction = 0x%X\n", Dio3GetDirection());

      printf("Current DIO status = 0x%X\n", Dio3GetInput());

      printf("Set DIO output to high\n");
      Dio3SetOutput(0x0F);

      printf("Set DIO output to low\n");
      Dio3SetOutput(0x00);
*/

      //for GPIO30..37
      Dio3SetDirection(0xF0); //GP30..33 = input, GP34..37=output

      Dio3SetOutput(0x00);                  //clear
//      DIO = Dio3GetInput() & 0x0F;
```

```
        Dio3SetOutput(0x00);               //clear
        DIO = Dio3GetInput() & 0x0F;
        if (DIO != 0x0A)
        {
              printf("The Fintek 81866 digital IO abnormal, abort.\n");
              return(1);
        }//if (DIO != 0x0A)

        Dio3SetOutput(0xA0);               //clr# is high
        Dio3SetOutput(0xF0);               //clk and clr# is high
        Dio3SetOutput(0xA0);               //clr# is high

        DIO = Dio3GetInput() & 0x0F;
        if (DIO != 0x05)
        {
              printf("The Fintek 81866 digital IO abnormal, abort.\n");
              return(1);
        }
        printf("!!! Pass !!!\n");
        return 0;
}
//--------------------------------------------------------------------------
void Dio3Initial(void)
{
        unsigned char ucBuf;

        // Switch GPIO multi-function pin for gpio 30~37
        // Multi Pin Select UART2/GPIOs
        // bit[6:5] 0:UART2 / 1:GPIOs

        ucBuf = Get_F81866_Reg(0x28);
        ucBuf |= BIT5;
        Set_F81866_Reg(0x28, ucBuf);

        Set_F81866_LD(0x06);                              //switch to logic device 6

        //enable the GP3 group
        ucBuf = Get_F81866_Reg(0x30);
        ucBuf |= 0x01;
        Set_F81866_Reg(0x30, ucBuf);

        Set_F81866_Reg(0x88, 0x00);                       //define as input mode
        //Set_F81866_Reg(0xA3, 0xFF);                             //push pull mode
}
//--------------------------------------------------------------------------
void Dio3SetOutput(unsigned char NewData)
{
        Set_F81866_LD(0x06);                              //switch to logic device 6
        Set_F81866_Reg(0x89, NewData);
}
//--------------------------------------------------------------------------
unsigned char Dio3GetInput(void)
{
        unsigned char result;

        Set_F81866_LD(0x06);                              //switch to logic device 6
        result = Get_F81866_Reg(0x8A);
        return (result);
}
```

```
//-------------------------------------------------------------------------
void Dio3SetDirection(unsigned char NewData)
{
        //NewData : 1 for input, 0 for output
        Set_F81866_LD(0x06);                                    //switch to logic device 6
        Set_F81866_Reg(0x88, NewData);
}
//-------------------------------------------------------------------------
unsigned char Dio3GetDirection(void)
{
        unsigned char result;

        Set_F81866_LD(0x06);                                    //switch to logic device 6
        result = Get_F81866_Reg(0x88);
        return (result);
}
//-------------------------------------------------------------------------
```

Files of the F81866.CPP

```
//-------------------------------------------------------------------------
#include "F81866.H"
#include <dos.h>
//-------------------------------------------------------------------------
unsigned int F81866_BASE;
void Unlock_F81866 (void);
void Lock_F81866 (void);
//-------------------------------------------------------------------------
unsigned int Init_F81866(void)
{
       unsigned int result;
       unsigned char ucDid;

       F81866_BASE = 0x4E;
       result = F81866_BASE;

       ucDid = Get_F81866_Reg(0x20);
       if (ucDid == 0x10)                          //Fintek 81801
       {       goto Init_Finish;    }

       F81866_BASE = 0x2E;
       result = F81866_BASE;

       ucDid = Get_F81866_Reg(0x20);
       if (ucDid == 0x07)                          //Fintek 81801
       {       goto Init_Finish;    }

       F81866_BASE = 0x00;
       result = F81866_BASE;

Init_Finish:
       return (result);
}
//----------------------------------------------------------------------
void Unlock_F81866 (void)
{
       outportb(F81866_INDEX_PORT, F81866_UNLOCK);
       outportb(F81866_INDEX_PORT, F81866_UNLOCK);
}
//----------------------------------------------------------------------
void Lock_F81866 (void)
{
       outportb(F81866_INDEX_PORT, F81866_LOCK);
}
//----------------------------------------------------------------------
void Set_F81866_LD( unsigned char LD)
{
       Unlock_F81866();
       outportb(F81866_INDEX_PORT, F81866_REG_LD);
       outportb(F81866_DATA_PORT, LD);
       Lock_F81866();
}
//----------------------------------------------------------------------
void Set_F81866_Reg( unsigned char REG, unsigned char DATA)
{
       Unlock_F81866();
       outportb(F81866_INDEX_PORT, REG);
       outportb(F81866_DATA_PORT, DATA);
       Lock_F81866();
```

```
}
//------------------------------------------------------------------------
unsigned char Get_F81866_Reg(unsigned char REG)
{
        unsigned char Result;
        Unlock_F81866();
        outportb(F81866_INDEX_PORT, REG);
        Result = inportb(F81866_DATA_PORT);
        Lock_F81866();
        return Result;
}
//------------------------------------------------------------------------
```

Files of the F81866.H

```
//-------------------------------------------------------------------------
#ifndef __F81866_H
#define __F81866_H                       1
//-------------------------------------------------------------------------
#define      F81866_INDEX_PORT        (F81866_BASE)
#define      F81866_DATA_PORT         (F81866_BASE+1)
//-------------------------------------------------------------------------
#define      F81866_REG_LD            0x07
//-------------------------------------------------------------------------
#define F81866_UNLOCK                 0x87
#define      F81866_LOCK                      0xAA
//-------------------------------------------------------------------------
unsigned int Init_F81866(void);
void Set_F81866_LD( unsigned char);
void Set_F81866_Reg( unsigned char, unsigned char);
unsigned char Get_F81866_Reg( unsigned char);
//-------------------------------------------------------------------------
#endif          //__F81866_H
```