TKS-G20-LN05

Compact Embedded Controller

Intel® Atom™ D510 1.66GHz Processor

Dual LAN, 2/6 USB2.0, 2/6 COM, 1 VGA

2 PCI-Express Mini Card

TKS-G20-LN05 Manual 1st Ed.
May 2011

Copyright Notice

Acknowledgments

All other products' name or trademarks are properties of their respective owners.

- AMI is a trademark of American Megatrends Inc.
- CompactFlash™ is a trademark of the Compact Flash Association.
- Microsoft Windows® is a registered trademark of Microsoft Corp.
- Intel®, Atom™ are trademarks of Intel Corporation.
- PC/AT, PS/2, and VGA are trademarks of International Business Machines Corporation.

All other product names or trademarks are properties of their respective owners.

Packing List

Before you begin operating your PC, please make sure that the following materials are enclosed:

- 1 TKS-G20-LN05 Embedded Controller
- 1 CD-ROM for manual (in PDF format) and drivers

If any of these items should be missing or damaged, please contact your distributor or sales representative immediately.

Safety & Warranty

1. Read these safety instructions carefully.
2. Keep this user's manual for later reference.
3. Disconnect this equipment from any AC outlet before cleaning. Do not use liquid or spray detergents for cleaning. Use a damp cloth.
4. For pluggable equipment, the power outlet must be installed near the equipment and must be easily accessible.
5. Keep this equipment away from humidity.
6. Put this equipment on a firm surface during installation. Dropping it or letting it fall could cause damage.
7. The openings on the enclosure are for air convection. Protect the equipment from overheating. DO NOT COVER THE OPENINGS.
8. Make sure the voltage of the power source is correct before connecting the equipment to the power outlet.
9. Position the power cord so that people cannot step on it. Do not place anything over the power cord.
10. All cautions and warnings on the equipment should be noted.
11. If the equipment is not used for a long time, disconnect it from the power source to avoid damage by transient over-voltage.
12. Never pour any liquid into an opening. This could cause fire or electrical shock.
13. Never open the equipment. For safety reasons, only qualified service personnel should open the equipment.
14. If any of the following situations arises, get the equipment checked by service personnel:
    a. The power cord or plug is damaged.
    b. Liquid has penetrated into the equipment.
    c. The equipment has been exposed to moisture.

d. The equipment does not work well, or you cannot get it to work according to the user's manual.

e. The equipment has been dropped and damaged.

f. The equipment has obvious signs of breakage.

15. DO NOT LEAVE THIS EQUIPMENT IN AN ENVIRONMENT WHERE THE STORAGE TEMPERATURE IS BELOW -20°C (-4°F) OR ABOVE 55°C (131°F). IT MAY DAMAGE THE EQUIPMENT.

FCC

This device complies with Part 15 FCC Rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received including interference that may cause undesired operation.

Caution:
There is a danger of explosion if the battery is incorrectly replaced. Replace only with the same or equivalent type recommended by the manufacturer. Dispose of used batteries according to the manufacturer's instructions and your local government's recycling or disposal directives.

Below Table for China RoHS Requirements
产品中有毒有害物质或元素名称及含量

AAEON Boxer/ Industrial System

部件名称	有毒有害物质或元素					
	铅 (Pb)	汞 (Hg)	镉 (Cd)	六价铬 (Cr(VI))	多溴联苯 (PBB)	多溴二苯醚 (PBDE)
印刷电路板及其电子组件	×	○	○	○	○	○
外部信号连接器及线材	×	○	○	○	○	○
外壳	×	○	○	○	○	○
中央处理器与内存	×	○	○	○	○	○
硬盘	×	○	○	○	○	○
电源	×	○	○	○	○	○

O：表示该有毒有害物质在该部件所有均质材料中的含量均在 **SJ/T 11363-2006** 标准规定的限量要求以下。

X：表示该有毒有害物质至少在该部件的某一均质材料中的含量超出 **SJ/T 11363-2006** 标准规定的限量要求。

备注：
一、此产品所标示之环保使用期限，系指在一般正常使用状况下。
二、上述部件物质中央处理器、内存、硬盘、电源为选购品。

Appendix A Programming The Watchdog Timer

Appendix B DIO

Chapter

1

General Information

1.1 Introduction

The newest Boxer series TKS-G20-LN05 has been introduced by AAEON and it utilizes Intel® Atom™ processor. In this era of information explosion, the advertising of consumer products will not be confined to the family television, but will also spread to high-traffic public areas, like department stores, the bus, transportation station, the supermarket etc. The advertising marketing industry will resort to every conceivable means to transmit product information to consumers. System integrators will need a multifunction device to satisfy commercial needs for such public advertising.

The TKS-G20-LN05 is designed for indoor environments due to the following reasons; first, the TKS-G20-LN05 offers low power consumption system that while operating in ambient temperatures ranging from 0° to 45°C. The MTBF (Mean Time Before Failure) rating states that the TKS-G20-LN05 can operate up to 50,000 hours at 40°C ambient temperature, which indicates its careful and long-life design.

The TKS-G20-LN05 is a standalone high performance controller designed for long-life operation and with high reliability. It can replace traditional methods and become the mainstream controller for the multimedia entertainment market.

1.2 Features

- Intel® Atom™ D510 1.66 GHz Processor
- Intel ICH8M Chipse
- USB2.0 x 6, COM x 6, Compact Flash™ x 1, 8-Bit Digital I/O
- Gigabit Ethernet x 2
- 2CH HD Audio
- PCI-Express Mini Card x 2
- WiFi & Bluetooth Combo Module (Optional To Occupy One PCIe Slot)
- ATX/ACPI Power Mode
- Fanless Cooling System

1.3 Specifications

System

CPU	Intel® Atom™ D510 1.66 GHz Processor
Memory	DDR2 533/667 SODIMM x 1, Max. 2 GB
VGA	VGA x 1
Keyboard/Mouse	Keyboard & Mouse by USB
Ethernet	10/100/1000Base-TX Ethernet, RJ-45 connector x 2
Hard Disk Storage	2.5" SATA HDD Bay x 1
Expansion	Mini Card Slot x 2
LCD/CRT Controller	Integrated in Processor, shared system memory up to 384MB
Solid Storage Disk	Type 2 CompactFlash™ slot x 1
Serial Port	RS-232/422/485 x 1, RS-232 x 5
USB	USB 2.0 x 6
System Control	Power ON/OFF
LED Indicator	Power LED x 1, HDD active LED x 1
Watchdog Timer	Generates a time-out system reset, setting via software
Others	WiFi & BT combo module through Mini PCIe slot (optional)
Power Supply	DC power adaptor input 12V
OS Support	WinXP, Win 7, Win Embedded, Win CE, Linux Fedora Core 13

Mechanical and Environmental

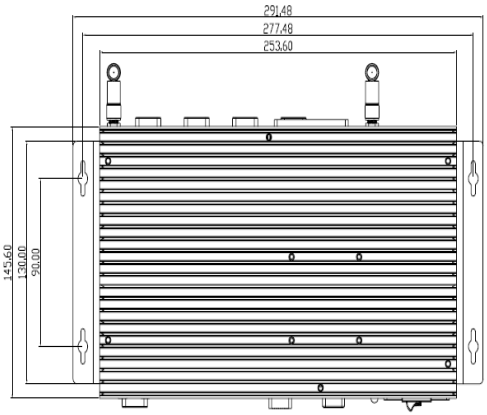
● Construction	Heavy Duty Steel Chassis
● Color	White
● Mounting	Wallmount
● Dimension	10" (D)x5.75"(W)x2.48"(H) (254mmx146mmx63mm)
● Gross Weight	5.33 lb (2.42 kg)
● Operating Temperature	32℉~113℉(0℃ ~ 45℃)
● Storage Temperature	-40°F ~ 176°F (-40°C ~ 80°C)
● Vibration	5 g rms/ 5~500Hz/ random operation –CFD 0.5 g rms/ 5~500Hz/ random operation –HDD
● Shock	50 G peak acceleration (11msec. duration) –CFD 10 G peak acceleration (11msec. duration) –HDD
● EMC	CE/FCC Class A

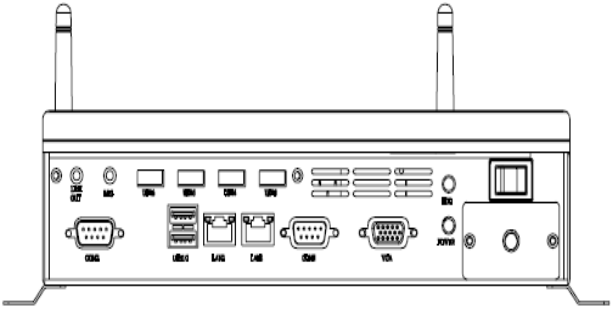
Chapter

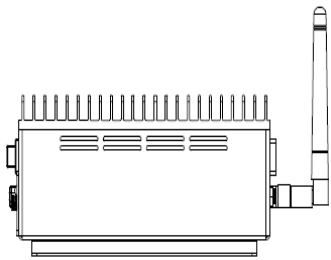
2

Quick Installation Guide

2.1 Dimension and I/O of TKS-G20-LN05

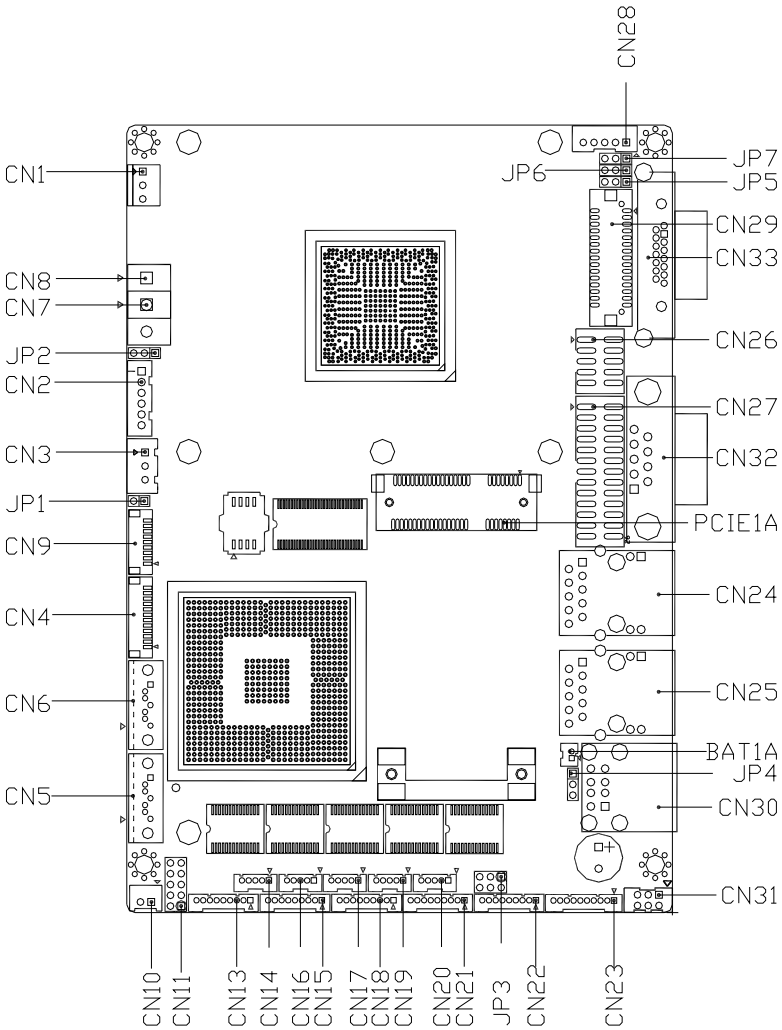




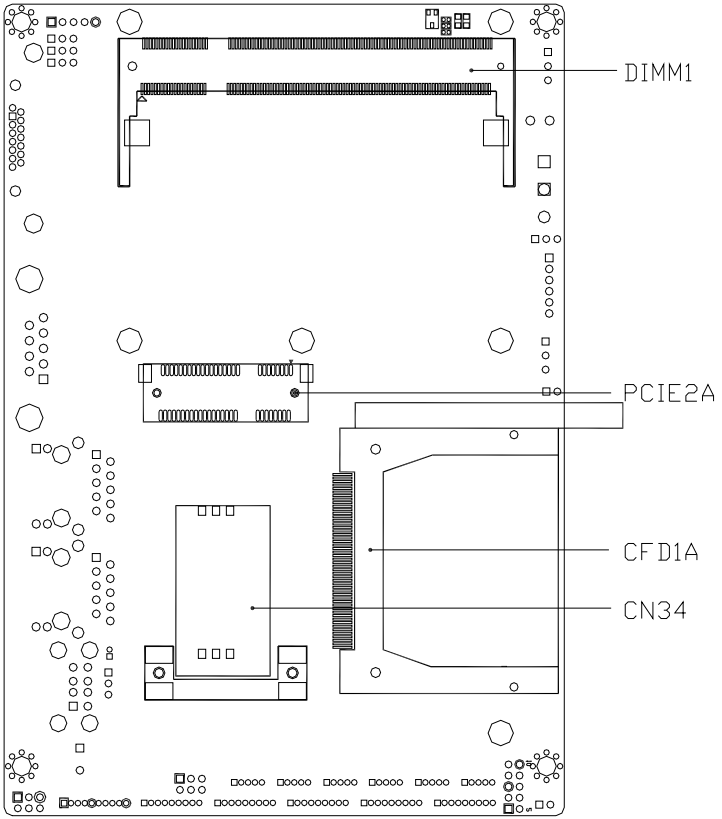


2.2 Location of Connectors and Jumpers of the Main Board

Component Side



Solder Side



2.3 List of Jumpers

The board has a number of jumpers that allow you to configure your system to suit your application.

The table below shows the function of each of the board's jumpers:

Label	Function
JP1	Touch Screen 4/5/8-wire Mode Selection
JP2	AT/ATX Power Mode Selection
JP3	COM2 RI/+5/+12V Selection
JP4	Clear CMOS
JP5	LVDS Inverter/ Backlight Bias/PWM Mode Selection
JP6	LVDS Operating Voltage Selection
JP7	LVDS Inverter/ Backlight Voltage Selection

2.4 List of Connectors

The board has a number of connectors that allow you to configure your system to suit your application.

The table below shows the function of each of the board's connectors:

Label	Function
CN1	CPU FAN
CN2	+5VSB Output w/ SMBus
CN3	External +5VSB Input
CN4	LPC Expansion I/F
CN5	SATA Port #1
CN6	SATA Port #2
CN7	External 5V Input (depend on power input configuration)
CN8	External 12V Input (depend on power input configuration)
CN9	Touch Screen Connector
CN10	+5V Output for SATA HDD using
CN11	Front Panel
CN13	COM Port #6
CN14	USB Port #7
CN15	COM Port #5
CN16	USB Port #6
CN17	USB Port #5
CN18	COM Port #4
CN19	USB Port #4
CN20	USB Port #3

CN21	COM Port #3
CN22	COM Port #2
CN23	Audio Line In/Out and MIC Connector
CN24	RJ-45 Ethernet #2
CN25	RJ-45 Ethernet #1
CN26	Digital I/O
CN27	Parallel Port
CN28	LVDS Inverter/ Backlight Connector
CN29	18-bit LVDS Output
CN30	USB Port #1 and #2
CN31	PS/2 Keyboard & Mouse
CN32	COM Port #1
CN33	Analog CRT Display
CN34	SIM Card Socket
CFD1	Compact Flash Disk
PCIE1	Mini-Card Slot #1
PCIE2	Mini-Card Slot #2
DIMM1	DDR2 SODIMM Slot

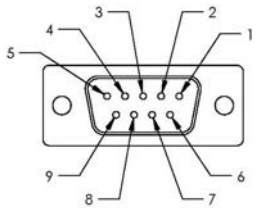
2.5 COM2 RS-232/422/485 Selection (CN22)

COM2 RS-232/422/485 selection for AAEON TKS series is set in BIOS setting as following:

Entering BIOS Setting Menu: Choose "Integrated Peripherals→ Super IO device →COM2 select". (Default setting is at "RS-232")

2.6 COM Port #2 RS-232/422/485 Port Connector (CN22)

Different devices implement the RS-232/422/485 standard in different ways. If you have problems with a serial device, check the pin assignments below for the connector.



RS-232 Mode

Pin	Signal	Pin	Signal
1	DCDB	2	DSRB
3	RXB	4	RTSB
5	TXB	6	CTSB
7	DTRB	8	RIB / +5 Volt. / (+12 Volt.)
9	Ground	10	N/C

RS-422 Mode

Pin	Signal	Pin	Signal
1	TXD-	2	N/C

3	RXD+	4	N/C
5	TXD+	6	N/C
7	RXD-	8	N/C / +5 Volt. / (+12 Volt.)
9	Ground	10	N/C

RS-485 Mode

Pin	Signal	Pin	Signal
1	TXD-	2	N/C
3	N/C	4	N/C
5	TXD+	6	N/C
7	N/C	8	N/C / +5 Volt. / (+12 Volt.)
9	Ground	10	N/C

2.7 Digital I/O Connector (CN26)

This connector offers 4-pair of digital I/O function.

BIOS using the I2C Bus to read/write internal DIO registers and the Serial Bus address is 0x6E.

The pin definitions are illustrated below:

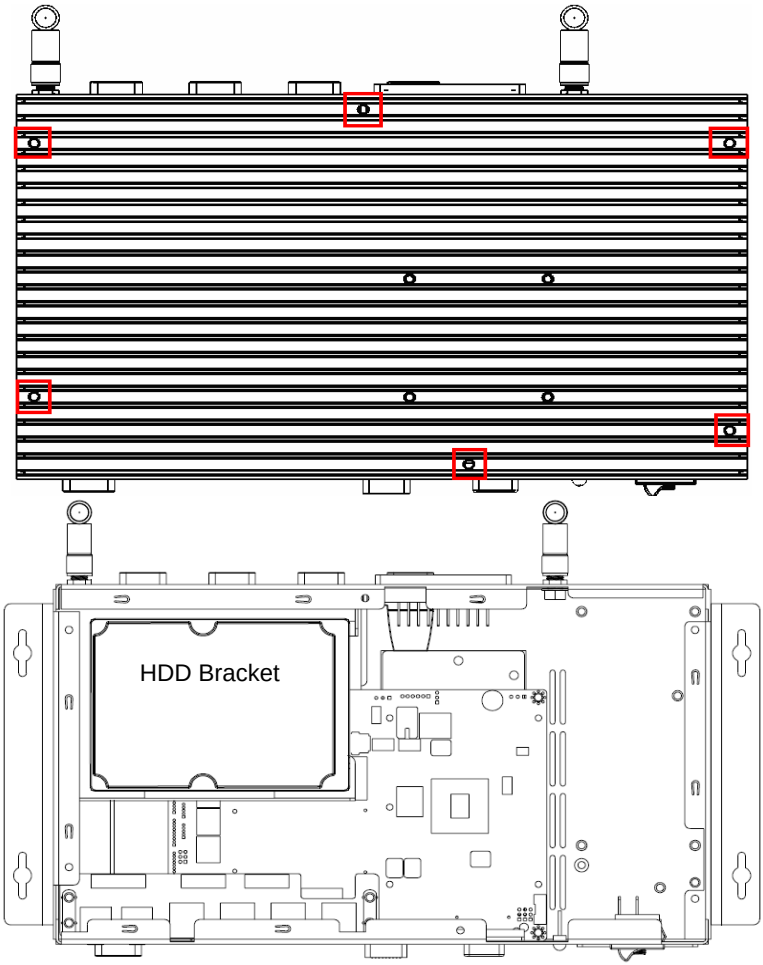
Pin	Signal	Pin	Signal
1	DIO_IN0	2	DIO_IN1
3	DIO_IN2	4	DIO_IN3
5	DIO_OUT0	6	DIO_OUT1
7	DIO_OUT2	8	DIO_OUT3
9	+3.3 Volt.	10	Ground

Note: The max. rating of Pin 1 ~ Pin 8 is 3.3V@8mA
The max. rating of Pin 9 is 3.3V@0.5A

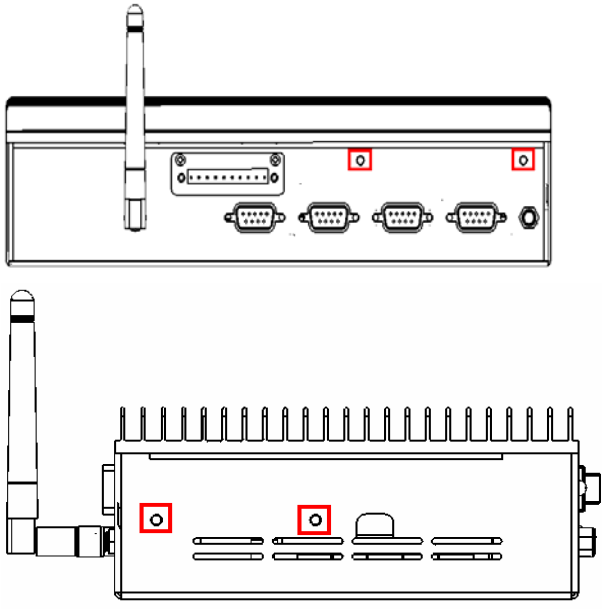
BIOS Setting (I2C address)	Connector Definition	Address(Register)		F75111 GPIO Setting
		Output	Input	
Port 1 @6Eh	Pin 1	21h/Bit 0	22h/Bit 0	U67 Pin 6 (GPIO 20)
Port 2 @6Eh	Pin 2	21h/Bit 1	22h/Bit 1	U67 Pin 7 (GPIO 21)
Port 3 @6Eh	Pin 3	21h/Bit 2	22h/Bit 2	U67 Pin 8 (GPIO 22)
Port 4 @6Eh	Pin 4	21h/Bit 3	22h/Bit 3	U67 Pin 24(GPIO 23)
Port 5 @6Eh	Pin 5	21h/Bit 4	22h/Bit 4	U67 Pin 23(GPIO 24)
Port 6 @6Eh	Pin 6	21h/Bit 5	22h/Bit 5	U67 Pin 22(GPIO 25)
Port 7 @6Eh	Pin 7	21h/Bit 6	22h/Bit 6	U67 Pin 21(GPIO 26)
Port 8 @6Eh	Pin 8	21h/Bit 7	22h/Bit 7	U67 Pin 20(GPIO 27)

2.8 Hard Disk Installation

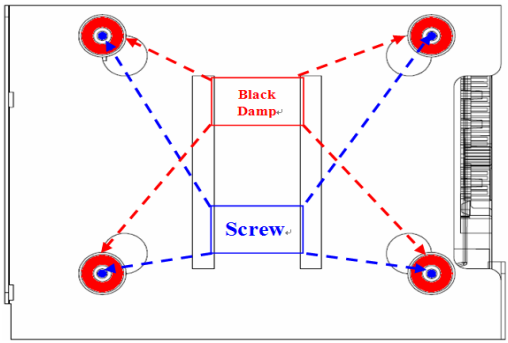
Step 1: Unfasten the screws on the top of the heat-sink and you will see the inside of the system.



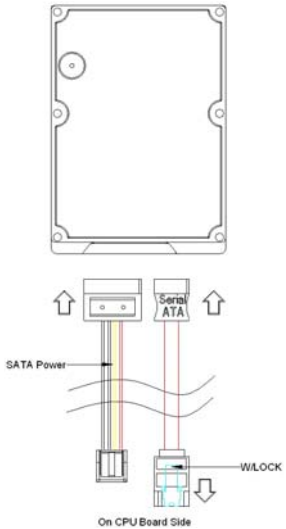
Step 2: Unfasten the two screws on the front side and right side of the chassis.



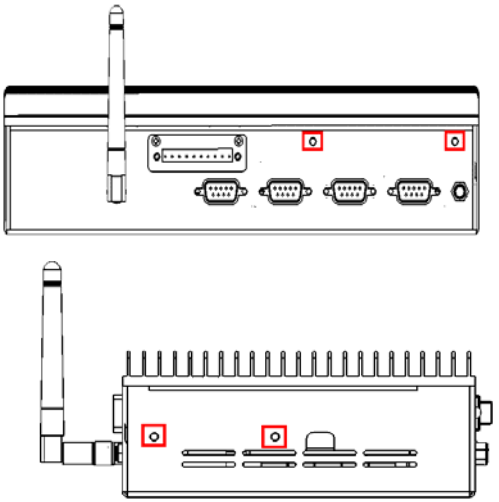
Step 3: Fasten the four HDD screws and black damper and then you can put the HDD in it.



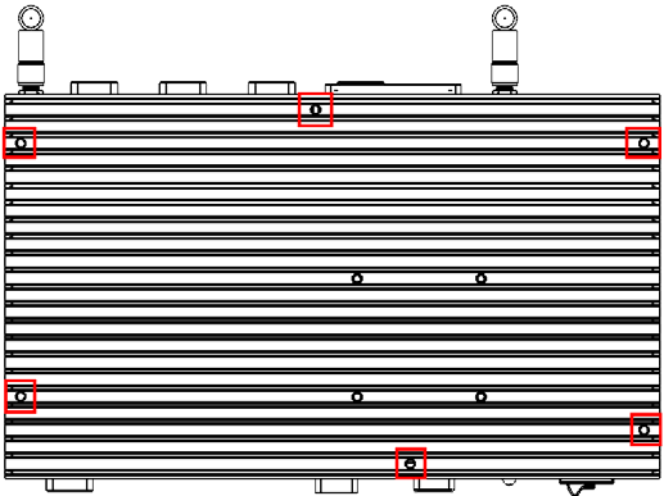
Step 4: Connect the HDD cables



Step 5: Fasten the two screws on the front side and right side of the chassis.

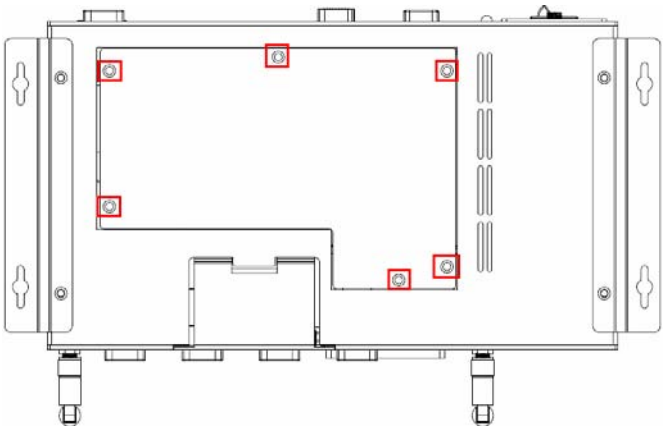


Step 6: Fasten the screws on the top of the heatsink.

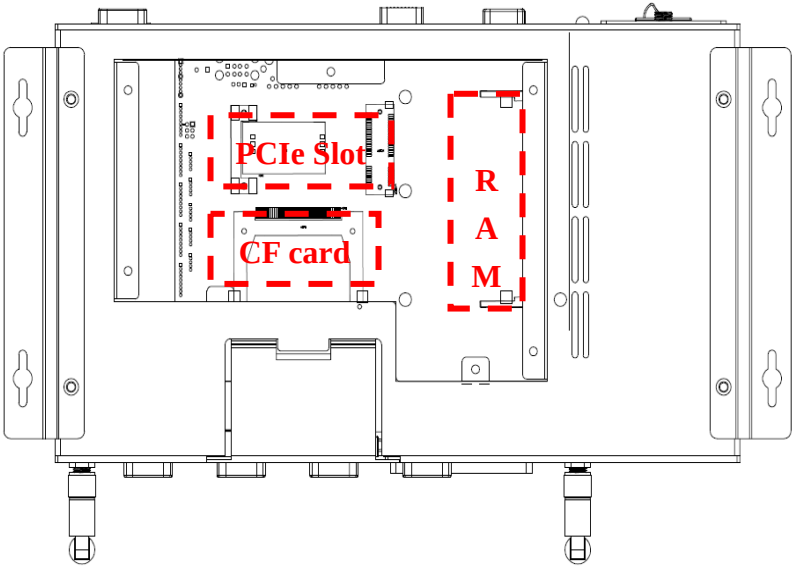


2.9 Accessory Installation

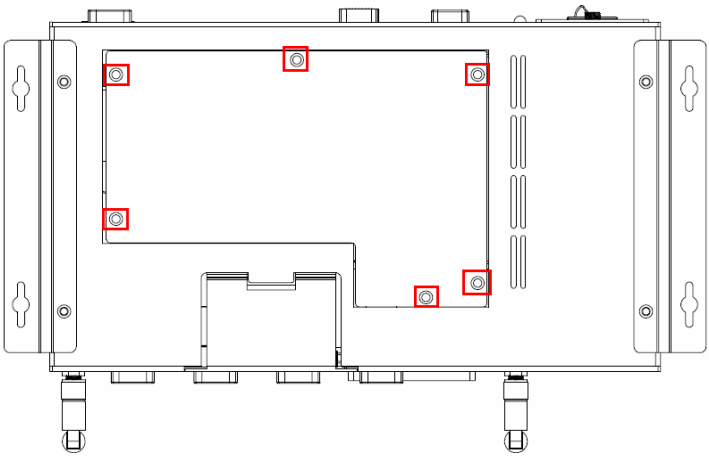
Step 1: Unfasten the screws on the rear panel



Step 2: You can see the inside placement of RAM, CF card, PCIe slot for you installation.

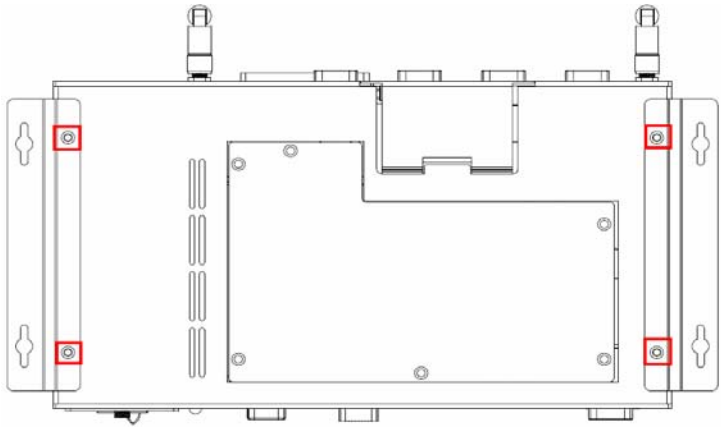


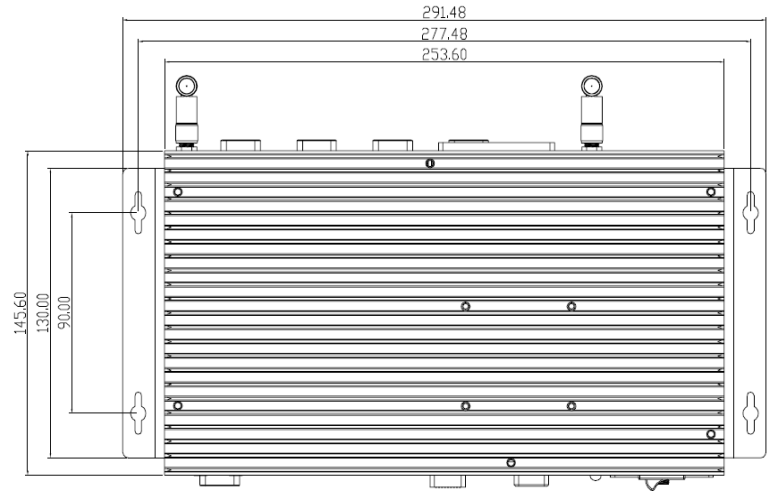
Step 3: Fasten the screws on the rear panel.



2.10 Wallmount Kit Installation

Get the brackets ready and fasten appropriate four screws on each bracket. After fastening the two brackets on the bottom lid of, the wall mount kit installation has been finished.





Chapter

3

AMI BIOS Setup

3.1 System Test and Initialization

These routines test and initialize board hardware. If the routines encounter an error during the tests, you will either hear a few short beeps or see an error message on the screen. There are two kinds of errors: fatal and non-fatal. The system can usually continue the boot up sequence with non-fatal errors.

System configuration verification

These routines check the current system configuration against the values stored in the CMOS memory. If they do not match, the program outputs an error message. You will then need to run the BIOS setup program to set the configuration information in memory.

There are three situations in which you will need to change the CMOS settings:

1. You are starting your system for the first time
2. You have changed the hardware attached to your system
3. The CMOS memory has lost power and the configuration information has been erased.

The TKS-G20-LN05 CMOS memory has an integral lithium battery backup for data retention. However, you will need to replace the complete unit when it finally runs down.

3.2 AMI BIOS Setup

AMI BIOS ROM has a built-in Setup program that allows users to modify the basic system configuration. This type of information is stored in battery-backed CMOS RAM so that it retains the Setup information when the power is turned off.

Entering Setup
Power on the computer and press <Del> or <F2> immediately. This will allow you to enter Setup.

Main
Set the date, use tab to switch between date elements.

Advanced
Enable disable boot option for legacy network devices.

Chipset
Host bridge parameters.

Boot
Enables/disable quiet boot option.

Security
Set setup administrator password.

Save&Exit
Exit system setup after saving the changes.

Chapter

4

Driver Installation

The TKS-G20-LN05 comes with a CD-ROM that contains all drivers and utilities that meet your needs.

Follow the sequence below to install the drivers:

Step 1 – Install Chipset Driver

Step 2 – Install VGA Driver

Step 3 – Install LAN Driver

Step 4 – Install Audio Driver

Step 5 – Install Wireless LAN Driver (Optional)

4.1 Installation:

Insert the TKS-G20-LN05 CD-ROM into the CD-ROM Drive. And install the drivers from Step 1 to Step 5 in order.

Step 1 – Install Chipset Driver

1. Click on the ***STEP1-CHIPSET*** folder and select the OS folder your system is
2. Double click on the ***infinst_autol.exe*** located in each OS folder
3. Follow the instructions that the window shows
4. The system will help you install the driver automatically

Step 2 – Install VGA Driver

1. Click on the ***STEP2-VGA*** folder and select the OS folder your system is
2. Double click on the ***Setup.exe*** located in each OS folder
3. Follow the instructions that the window shows
4. The system will help you install the driver automatically

Step 3 – Install LAN Driver

1. Click on the ***STEP3-LAN*** folder and select the OS folder your system is
2. Double click on the ***PROWin32.exe*** located in each OS folder
3. Follow the instructions that the window shows
4. The system will help you install the driver automatically

Step 4 – Install Audio Driver

1. Click on the ***STEP4-AUDIO*** folder and select the OS folder your system is
2. Double click on the ***Setup.exe*** located in each OS folder
3. Follow the instructions that the window shows
4. The system will help you install the driver automatically

Step 5 – Install Wireless LAN Driver (Optional)

1. Click on the ***STEP5-WIRELESS LAN*** folder and select the OS folder your system is
2. Select the folder of ***Install_CD***, and double click on the ***setup.exe*** located in each OS folder
3. Follow the instructions that the window shows
4. The system will help you install the driver automatically

Appendix

A

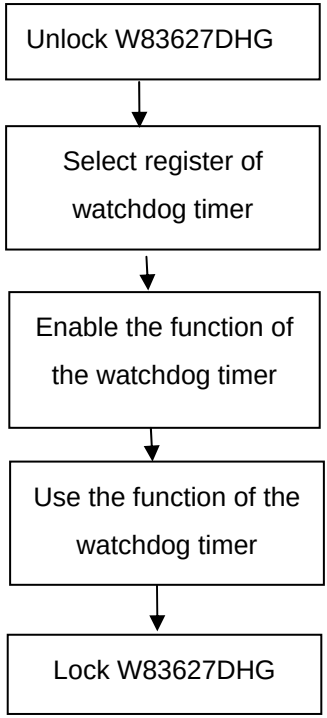
Programming the Watchdog Timer

A.1 Programming

TKS-G20-LN05 utilizes W83627DHG-P chipset as its watchdog timer controller.

Below are the procedures to complete its configuration and the AAEON intial watchdog timer program is also attached based on which you can develop customized program to fit your application.

Configuring Sequence Description



There are three steps to complete the configuration setup:

(1) Enter the W83627DHG config Mode

(2) Modify the data of configuration registers

(3) Exit the W83627DHG config Mode. Undesired result may occur if the config Mode is not exited normally.

(1) Enter the W83627DHG config Mode

To enter the W83627DHG config Mode, two special I/O write operations are to be performed during Wait for Key state. To ensure the initial state of the key-check logic, it is necessary to perform two write operations to the Special Address port (2EH). The different enter keys are provided to select configuration ports (2Eh/2Fh) of the next step.

	Address Port	Data Port
87h,87h:	2Eh	2Fh

(2) Modify the Data of the Registers

All configuration registers can be accessed after entering the config Mode. Before accessing a selected register, the content of Index 07h must be changed to the LDN to which the register belongs, except some Global registers.

(3) Exit the W83627DHG config Mode

The exit key is provided to select configuration ports (2Eh/2Fh) of the next step.

	Address Port	Data Port
0aah:	2Eh	2Fh

WatchDog Timer Register I (Index=F5h, Default=00h)

CRF5 (PLED and KBC P20 Control Mode Register)

Bit 7-5 : select PLED mode

= 000 Power LED pin is driven high.

= 001 Power LED pin outputs 0.5Hz pulse with 50% duty cycle.

= 010 Power LED pin is driven low.

= 011 Power LED pin outputs 2Hz pulse with 50% duty cycle.

= 100 Power LED pin outputs 1Hz pulse with 50% duty cycle.

= 101 Power LED pin outputs 4Hz pulse with 50% duty cycle.

= 110 Power LED pin outputs 0.25Hz pulse with 50% duty cycle.

=111 Power LED pin outputs 0.25Hz pulse with 50% duty cycle..

Bit 4 : WDTO# count mode is 1000 times faster.

= 0 Disable.

= 1 Enable.

Bit 3 : select WDTO# count mode.

= 0 second

= 1 minute

Bit 2 : Enable the rising edge of keyboard Reset (P20) to force Time-out event.

= 0 Disable

= 1 Enable

Bit 1 : Disable / Enable the WDTO# output low pulse to the KBRST# pin (PIN60)

= 0 Disable

= 1 Enable

Bit 0 : Reserved.

WatchDog Timer Register II (Index=F6h, Default=00h)

Bit 7-0 = 0 x 00 Time-out Disable

= 0 x 01 Time-out occurs after 1 second/minute

= 0 x 02 Time-out occurs after 2 second/minutes

= 0 x 03 Time-out occurs after 3 second/minutes

……..

= 0 x FF Time-out occurs after 255 second/minutes

WatchDog Timer Register III (Index=F7h, Default=00h)

Bit 7 : Mouse interrupt reset Enable or Disable

= 1 Watchdog Timer is reset upon a Mouse interrupt

= 0 Watchdog Timer is not affected by Mouse interrupt

Bit 6 : Keyboard interrupt reset Enable or Disable

= 1 Watchdog Timer is reset upon a Keyboard interrupt

= 0 Watchdog Timer is not affected by Keyboard interrupt

Bit 5 : Force Watchdog Timer Time-out. Write Only

= 1 Force Watchdog Timer time-out event: this bit is self-clearing

Bit 4 : Watchdog Timer Status. R/W

= 1 Watchdog Timer time-out occurred

= 0 Watchdog Timer counting

Bit 3-0 : These bits select IRQ resource for Watchdog. Setting of 2 selects SMI.

A.2 W83627DHG Watchdog Timer Initial Program

Example: Setting 10 sec. as Watchdog timeout interval

```
;////////////////////////////////////////////////////////////////////////////////
Mov dx,2eh              ;Enter W83627DHG config mode
Mov al,87h              (out 87h to 2eh twice)
Out dx,al
Out dx,al
;////////////////////////////////////////////////////////////////////////////////
Mov al,07h
Out dx,al
Inc dx
Mov al,08h              ;Select Logical Device 8 (GPIO Port 2)
Out dx,al
;////////////////////////////////////////////////////////////////////////////////
Dec dx
Mov al,30h              ;CR30 (GP20~GP27)
Out dx,al
Inc dx
Mov al,01h              ;Activate GPIO2
Out dx,al
```

```
;////////////////////////////////////////////////////////////////////////////////////////
Dec dx
Mov al,0f5h             ;CRF5 (PLED mode register)
Out dx,al
Inc dx
In al,dx
And al,not 08h          ;Set second as counting unit
Out dx,al
;////////////////////////////////////////////////////////////////////////////////////////
Dec dx
Mov al,0f6h             ; CRF6
Out dx,al
Inc dx
Mov al,10               ;Set timeout interval as 10 sec.
Out dx,al
;////////////////////////////////////////////////////////////////////////////////////////
Dec dx                  ;Exit W83627DHG config mode
Mov al,0aah                     (out 0aah to 2eh once)
Out dx,al
;////////////////////////////////////////////////////////////////////////////////////////
```

Appendix

B

DIO

B.1 DIO

The F75111 provides one serial access interface, I2C Bus, to read/write internal registers. The address of Serial Bus is 0x6E (0110_1110)

The related register for configuring DIO is list as follows:

Configuration and Control Register – Index 01h

Power-on default [7:0]=0000_1000b

Bit	Name	R/W	PWR	Description
7	INIT	R/W	VSB3V	Software reset for all registers including Test Mode registers. Users use only.
6	Reserved	R/W	VSB3V	
5	EN_WDT10	R/W	VSB3V	Enable Reset Out. If set to 1, enable WDTOUT10# output. Default is disable.
4	Reserved	R/W	VSB3V	
3	Reserved	R/W	VSB3V	
2	Reserved	R/W	VSB3V	
1	SMART_POWR_MANAGEMENT	R/W	VSB3V	Set this bit to 1 will enable auto power down mode, when all function are idle then 20ms the chip will auto power down, it will wakeup when GPIO state change or read write register
0	SOFT_POWR_DOWN	R/W	VSB3V	Set this bit to 1 will power down all of the analog block and stop internal clock, write 0 to clear this bit or when GPIO state change will auto clear this bit to 0.

GPIO2x Output Control Register – Index 20h

Power-on default [7:0]=0000_1000b

Bit	Name	R/W	PWR	Description
7	GP27_OCTRL	R/W	VSB3V	GPIO 27 output control. Set to 1 for output function. Set to 0 for input function (default).
6	GP26_OCTRL	R/W	VSB3V	GPIO 26 output control. Set to 1 for output function. Set to 0 for input function (default).
5	GP25_OCTRL	R/W	VSB3V	GPIO 25 output control. Set to 1 for output function. Set to 0 for input function (default).
4	GP24_OCTRL	R/W	VSB3V	GPIO 24 output control. Set to 1 for output function. Set to 0 for input function (default).
3	GP23_OCTRL	R/W	VSB3V	GPIO 23 output control. Set to 1 for output function. Set to 0 for input function (default).
2	GP22_OCTRL	R/W	VSB3V	GPIO 22 output control. Set to 1 for output function. Set to 0 for input function (default).
1	GP21_OCTRL	R/W	VSB3V	GPIO 21 output control. Set to 1 for output function. Set to 0 for input function (default).
0	GP20_OCTRL	R/W	VSB3V	GPIO 20 output control. Set to 1 for output function. Set to 0 for input function (default).

GPIO2x Output Data Register – Index 21h

Power-on default [7:0]=0000_1000b

Bit	Name	R/W	PWR	Description
7	GP27_ODATA	R/W	VSB3V	GPIO 27 output data.
6	GP26_ODATA	R/W	VSB3V	GPIO 26 output data.
5	GP25_ODATA	R/W	VSB3V	GPIO 25 output data.
4	GP24_ODATA	R/W	VSB3V	GPIO 24 output data.
3	GP23_ODATA	R/W	VSB3V	GPIO 23 output data.
2	GP22_ODATA	R/W	VSB3V	GPIO 22 output data.
1	GP21_ODATA	R/W	VSB3V	GPIO 21 output data.
0	GP20_ODATA	R/W	VSB3V	GPIO 20 output data.

GPIO2x Input Status Register – Index 22h

Power-on default [7:0]=xxxx_xxxxb

Bit	Name	R/W	PWR	Description
7	GP27_PSTS	RO	VSB3V	Read the GPIO27 data on the pin.
6	GP26_PSTS	RO	VSB3V	Read the GPIO26 data on the pin.
5	GP25_PSTS	RO	VSB3V	Read the GPIO25 data on the pin.
4	GP24_PSTS	RO	VSB3V	Read the GPIO24 data on the pin.
3	GP23_PSTS	RO	VSB3V	Read the GPIO23 data on the pin.
2	GP22_PSTS	RO	VSB3V	Read the GPIO22 data on the pin.
1	GP21_PSTS	RO	VSB3V	Read the GPIO21 data on the pin.
0	GP20_PSTS	RO	VSB3V	Read the GPIO20 data on the pin.

The following is a sample code for “4 input 4 output read/write.”

```
#include <stdio.h>
#include <stdlib.h>
#include <dos.h>

#define smbase 0xF000
#define Show_Len 0x50
#define dev_addr 0x6E

#define SMBUS_REG_STATUS   0    //defined in SMBUS_STATUS_*
#define SMBUS_REG_COMMAND 2      //Write operation, defined in
SMBUS_CMD_*
#define SMBUS_REG_RESET_POINTER 2 //Read operation
#define SMBUS_REG_DATA_OFFSET  3    //b7:0 = Byte (Word) Offset
#define SMBUS_REG_DID_RW   4    //b7:1 = DeviceID, b0 = Read/Write

#define SMBUS_REG_HST_D0   5        //DATA 0 register
#define SMBUS_CMD_START             0x40
#define SMBUS_CMD_CMD_RW            0x08       //Command read/write

#define SMBUS_DATA_READ                         0x01
#define SMBUS_DATA_WRITE                0x00
#define SMBUS_STATUS_BYTE_DONE                  0x80
#define SMBUS_STATUS_IN_USE                     0x40
#define SMBUS_STATUS_SMBALERT           0x20
#define SMBUS_STATUS_FAILED                     0x10
```

```
#define SMBUS_STATUS_BUS_ERROR                  0x08
#define SMBUS_STATUS_DEVICE_ERROR    0x04
#define SMBUS_STATUS_INTR  0x02   //After read/write done
#define SMBUS_STATUS_HOST_BUSY                  0x01   //


int main(void)

        {

      int i,out_reg,in_reg,data_offset;
      int s_count;
      unsigned char s_data, w_data,f_data0,f_data1,f_data2,f_data3;

      printf("\n\t
#*#=============================================#*#\n");
      printf("\t |||       AAEON DIO Test Program For 4_In+4_Out      |||\n");
      printf("\t
#*#=============================================#*#\n");
      printf("\t ");

         out_reg=0x21;
         in_reg=0x22;

         //================Test_Pattern=0x0F==================
         //Write GP20-27=0Fh
                  SMBusWrite(smbase, dev_addr , out_reg, 0x0F);
         //Read GP20-27
                  f_data3=SMBusRead(smbase, dev_addr , in_reg);

         return 0;
}
/* send_i2c_address() */


int IO_Delay(int time)
{
```

```
        int     i;
        for(i=0;i<time;i++)
        {
                outportb(0xeb,0xFF);
        }
        return 0;
}


int SMBusGetStatus(unsigned int SMBus_Base)
{       //no error pending
        return inportb(SMBus_Base+SMBUS_REG_STATUS);
}

void SMBusClearStatus(unsigned int SMBus_Base)
{
        unsigned char status;
        do
        {
                status = inportb(SMBus_Base+SMBUS_REG_STATUS);
                outportb(SMBus_Base+SMBUS_REG_STATUS, status);
        }while( status );

}


int SMBusCheckReady(unsigned int SMBus_Base)
{
        int     i;
        unsigned char Org,Temp;

        for(i=0;i<2048;i++)
        {
        Temp = inportb(SMBus_Base+SMBUS_REG_STATUS);
        //get status
        IO_Delay(1);
      outportb(SMBus_Base+SMBUS_REG_STATUS, Temp);          //clear
```

```
status
        IO_Delay(1);

                if ((Temp & 0x02) == 1)
        //termination of command ?
                {
                return 0;
                }

                if ((Temp & 0x40) == 0)
        //status OK ?
                {
                return 0;
                }

                if ((Temp & 0x04) == 1)
        //device error
                {
                return -1;
                }
        }

        return -1;

}


int SMBusCheckComplete(unsigned int SMBus_Base)
{
        unsigned int result;
        result = inportb(SMBus_Base+SMBUS_REG_STATUS);

        if((result &
(SMBUS_STATUS_FAILED+SMBUS_STATUS_BUS_ERROR+SMBUS_S
TATUS_DEVICE_ERROR+SMBUS_STATUS_HOST_BUSY)) != 0)
                return 0;
        else
```

```
                return -1;
}


int SMBusRead(unsigned int SMBus_Base, int DeviceID, int
REG_DATA_OFFSET)
{
        unsigned char read_data;

//      outportb(SMBus_Base+SMBUS_REG_COMMAND, 0x00);
        //Clear previous commands
        SMBusCheckReady( SMBus_Base );

        outportb(SMBus_Base+SMBUS_REG_DID_RW, DeviceID +
SMBUS_DATA_READ);                       //ID cmd(read)
        IO_Delay(1);
        IO_Delay(1);
     SMBusClearStatus(SMBus_Base);
        outportb(SMBus_Base+SMBUS_REG_DATA_OFFSET,
REG_DATA_OFFSET);                               //Index
        IO_Delay(1);
        IO_Delay(1);
        outportb(SMBus_Base+SMBUS_REG_COMMAND,
SMBUS_CMD_START+SMBUS_CMD_CMD_RW);    //Read data
        IO_Delay(500);
        SMBusCheckReady( SMBus_Base );   //IO_Delay(1000);
        read_data = inportb(SMBus_Base+SMBUS_REG_HST_D0);
                                        //Data0
        IO_Delay(1);
        IO_Delay(1);

        return read_data;

}
int SMBusWrite(unsigned int SMBus_Base, int DeviceID, int
REG_DATA_OFFSET, int data)
{
```

```
        int     status;

        SMBusCheckReady( SMBus_Base );

        outportb(SMBus_Base+SMBUS_REG_DID_RW, DeviceID +
SMBUS_DATA_WRITE);          //ID cmd(Write)
        IO_Delay(1000);
        IO_Delay(1000);
      SMBusClearStatus(SMBus_Base);
        outportb(SMBus_Base+SMBUS_REG_DATA_OFFSET,
REG_DATA_OFFSET);                       //Index
        IO_Delay(1000);
        IO_Delay(1000);
        outportb(SMBus_Base+SMBUS_REG_HST_D0, data);
                                        //Data0
        IO_Delay(1000);
        IO_Delay(1000);
        outportb(SMBus_Base+SMBUS_REG_COMMAND,
SMBUS_CMD_START+SMBUS_CMD_CMD_RW);    //write data
        IO_Delay(1000);
        IO_Delay(1000);

        IO_Delay(100);
        status=SMBusCheckReady( SMBus_Base );

        if (status<0)
        {
                printf("\n Write Fail\n");
        }
        else
        {
//              printf("\n Write Success\n");
        }

        return 0;
}
```