Lanner

creating value in applied computing

In-Vehicle Computing

Hardware Platforms for mobile applications

LVC-2000
V1.0

User's Manual

Release Date: 2014/12/10

Overview

Icon Descriptions

The icons are used in the manual to serve as an indication of interest topics or important messages. Below is a description of these icons:

NOTE: This check mark indicates that there is a note of interest and is something that you should pay special attention to while using the product.

WARNING: This exclamation point indicates that there is a caution or warning and it is something that could damage your property or product.

Online Resources

The listed websites are links to the on-line product information and technical support.

Resource	Website
Lanner	http://www.lannerinc.com
Product Resources	http://assist.lannerinc.com
RMA	http://eRMA.lannerinc.com

Copyright and Trademarks

This document is copyrighted, © 2014. All rights are reserved. The original manufacturer reserves the right to make improvements to the products described in this manual at any time without notice.

No part of this manual may be reproduced, copied, translated or transmitted in any form or by any means without the prior written permission of the original manufacturer. Information provided in this manual is intended to be accurate and reliable. However, the original manufacturer assumes no responsibility for its use, nor for any infringements upon the rights of third parties that may result from such use.

Acknowledgement

Intel, Pentium and Celeron are registered trademarks of Intel Corp.

Microsoft Windows and MS-DOS are registered trademarks of Microsoft Corp.

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

Compliances and Certification

CE Certification

This product has passed the CE test for environmental specifications. Test conditions for passing included the equipment being operated within an industrial enclosure. In order to protect the product from being damaged by ESD (Electrostatic Discharge) and EMI leakage, we strongly recommend the use of CE-compliant industrial enclosure products.

FCC Class A Certification

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

e Mark Certification

E13 - Luxembourg

Mechanical compliance

Vibration:

- General Vibration (operating): Refer to MIL-STD-810G, Method 514.6, Procedure I (Transportation), Category 4 – Common carrier (US highway truck vibration exposure)
- General Vibration (non-operating): Refer to MIL-STD-810G, Method 514.6, Procedure I (Transportation), Category 24 – General minimal integrity

Shock:

- Operating (Functional Test for Ground Equipment): Refer to MIL-STD-810G, Method 516.6, Procedure I, 40g, 11ms
- B. Non-Operating (Crash Hazard Shock Test for Ground Equipment): Refer to MIL-STD-810G, Method 516.6, Procedure V, 75g, 11ms

Electrical transient conduction along supply lines only (12V/24V)

Revision History

0.1	2014/09/14	Preliminary
1.0	2014/12/10	Official release

Table of Contents

Chapter 1: Introduction

Thank you for choosing LVC-2000. The entry-level box PC is one of the most compact in-vehicle computing system which equips with a vibration kit to eliminate shock and vibration. It is designed to be installed on a moving transportation system.

LVC-2000 is a fanless in-vehicle computer with MIL-STD-810G certified shock and vibration resistance. Built with onboard Intel® Atom™ processor E3845 (codenamed "Bay Trail), the in-vehicle computer is a value time-to-market solution with enhanced performance and low power consumption. LVC-2000 also features multiple I/O connectivity including CAN bus (module optional), LAN port, GPS/G-sensor, COM ports, multiple Digital I/Os, and mini PCI Express sockets, making it perfect for vehicle monitoring, in-car infotainment and fleet management.

Features:

- Onboard Intel® Quad-core Bay Trail SoC
- Vehicle Power Ignition Management
- The MIO Connector
- MIL-STD-810G Certification for Shock & Vibration Resistance
- Fanless Design and Aluminum Enclosure
- Wide Operating Temperature Workability
- Support VGA, HDMI dual independent display interface.
- 2x mini-PCIe sockets with two support WiFi and 3G wireless connection.
- Suspending Kit or wall mounting
- Support 12V DC output @Max 1A
- Onboard Ublox NEO-7N GPS receiver module

System Specifications

Dimensions (WxDxH,)		198W x 165D x 52H (mm, the unit)
Processor		Intel® BayTrail E3845 1.91 GHz Optional for E3815 / E3825 / E3826 / E2827
System Memory	Module type	DDR3L SO-DIMM x1 (up to 8GB)
BIOS		AMI SPI Flash BIOS
Storage	mSATA/ SATA	1 x mSATA with SATA 3.0Gbps 1 x SATA 2.5" drive bay for HDD/ SSD
Ethernet Controller		Intel i210IT
Graphic Controller		Intel Integrated HD graphic engine
Audio Controller		Realtek ALC886-GR
Super I/O		1x LPC Super I/O Fintek F81865F supporting DIO, Serial ports, Watchdog Timer, Hardware monitor and Temperature meter for internal system
I/O	LAN	GbE RJ45 x1
	Display	VGA: up to 1600x1200@60 24bpp HDMI: up to 1920x1080 @60
	Audio	Internal pin header for Mic-in and Line-out
	Serial I/O	1 x DB9 RS-232/422/485 1 x DB9 RS-232/422/485 (RS-232 by default)
	GPS	Ublox NEO-7N GPS receiver
	G-sensor	ADXL 345
	MIO	4x DI (5V or 12V TTL selectable) 4x DO (12V TTL , Max. 100mA) 2x MCU DI 2x Relay 1x 12V Output @Max. 1A
	USB	USB 3.0 Type A x 1 USB 2.0 x 2 by internal pin header
	Power Input	3-pin terminal block (DC9-36V, GND, Ignition)
	Expansion	1 x full-size mini-PCIexpress socket (USB+PCIe) with SIM-card reader 1 x half-size mini-PCIexpress socket
	CAN bus	supports J1939 & J1708, (module optional)
Power Input		1x DC 9~36V, GND and Ignition
MCU		1x MICRO-CONTROLLER LPC1114FBD48/301 SMD PHILIP, Support 2xDI
Lanner Ignition System Management		Ignition Control Utility under Windows Base OS. Ignition Control Sample Code for Linux OS
OS Support	**Windows**	Driver Support: Windows 7/7 Embedded/8 embedded OS Image: W7 FES (64bit & 32bit) / Windows 8(32bit)
	Linux	Driver Support: Linux kernel 2.6.X or later

Thermal Solution		Fanless system. Heat dissipate from aluminum enclosure
Mounting		Suspending Kit or wall mounting-1
Certifications		CE, FCC Class A, E13, RoHS
Compliance		Vibration: MIL-STD-810G, Method 514.6 Shock:MIL-STD-810G, Method 516.6
Operating Temperature Range	Extended	-20~60°C (with industrial components)

Package Contents

Your package contains the following items:

LVC-2000 Fanless Embedded System with rubber stands:

- Terminal Block Connectors:

-Power connector 3 pin x1 (P/N:04AW20031E001)

-MIO Connector 20 pin x1 (P/N: 04AW20203Z101)

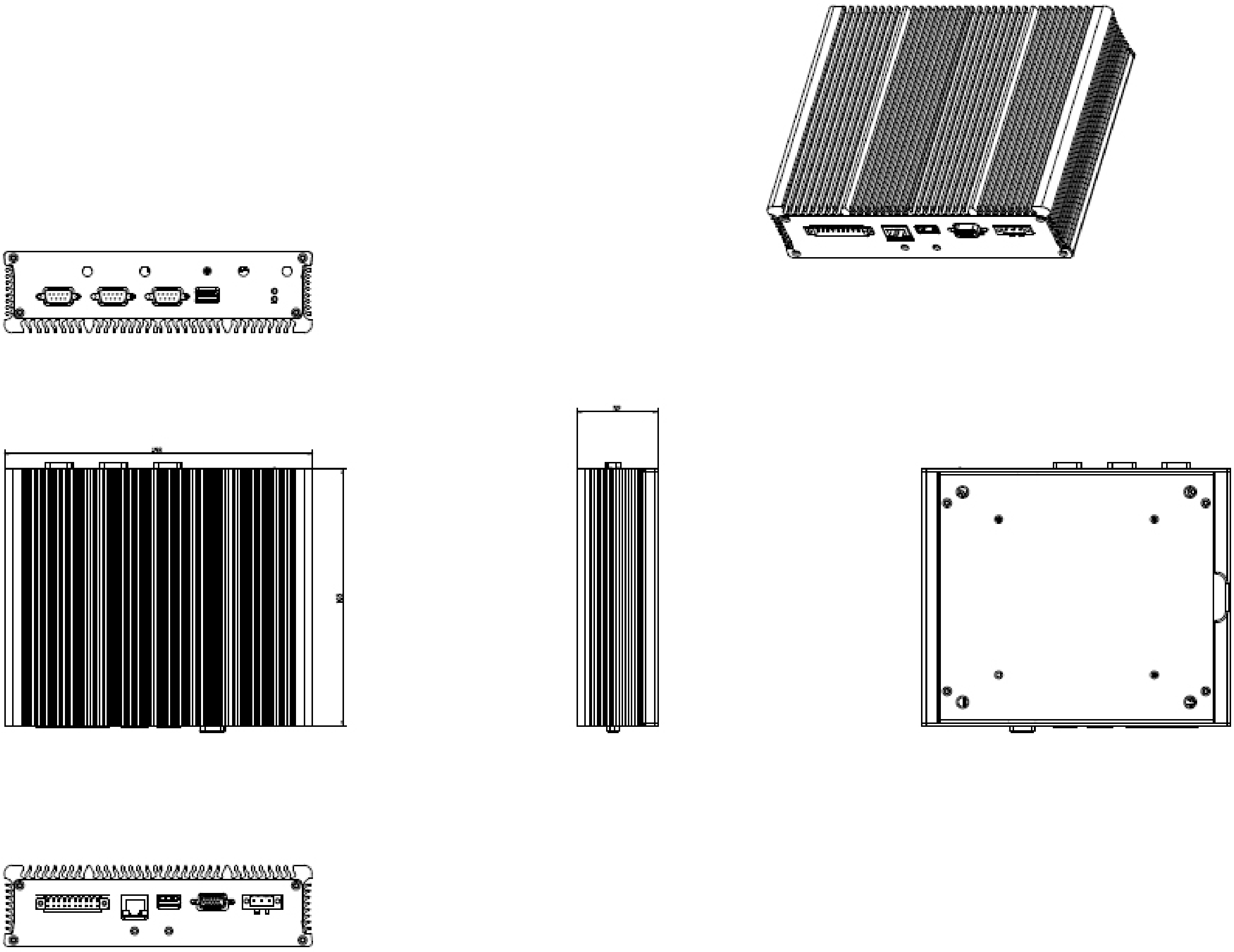
- HDD Screws x 4 (P/N: 070W102400602)
- Mini-PCIe Screws x 4 (P/N: 070W101000401)

Chapter 2:
System Components

Mechanical Drawings

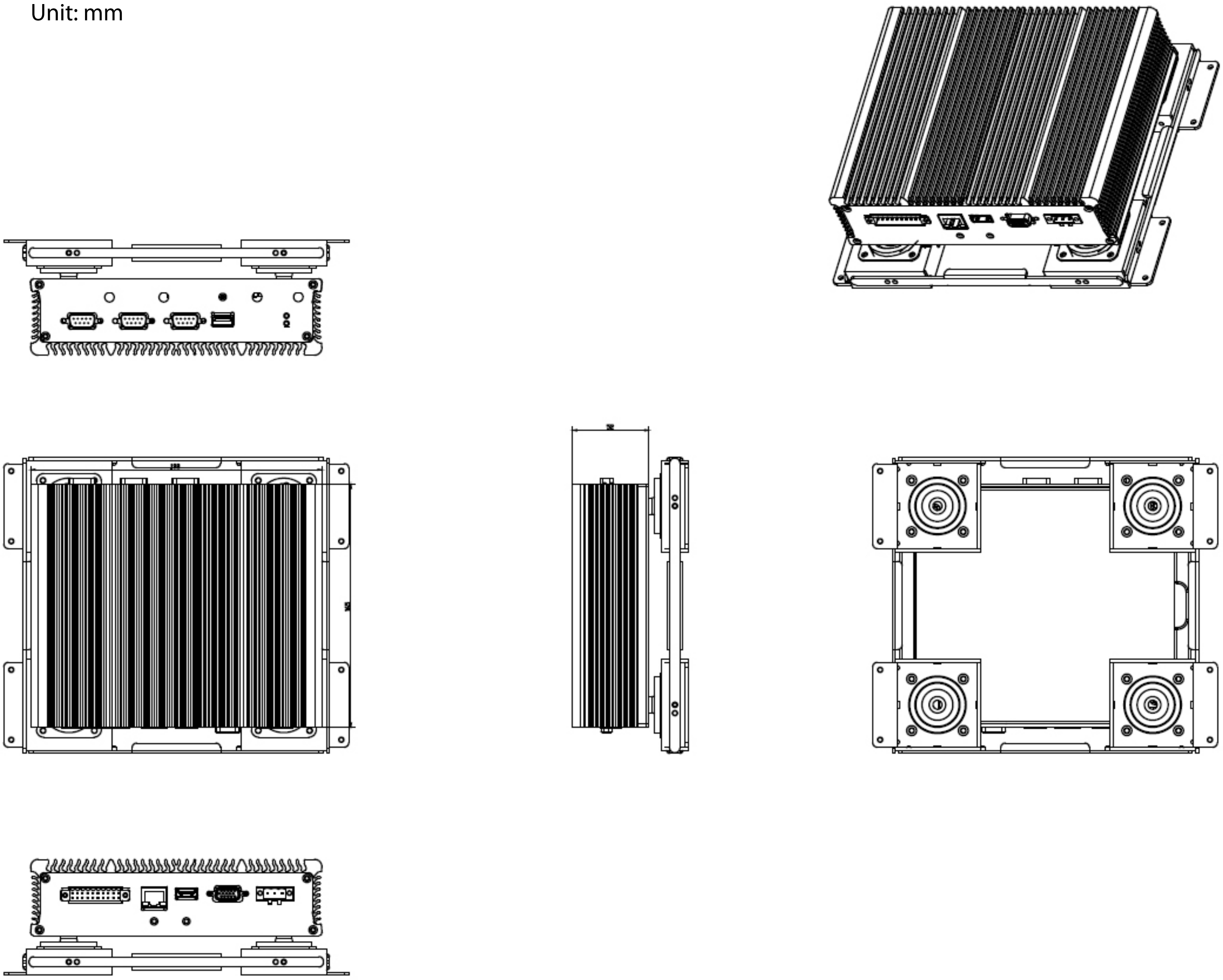
Mechanical dimensions of the LVC-2000 with the system itself

Unit: mm



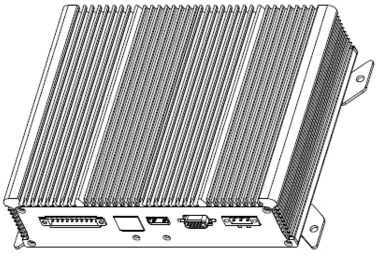
Mechanical dimensions of the LVC-2000 with anti-vibration kit

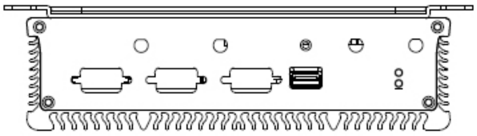
Unit: mm

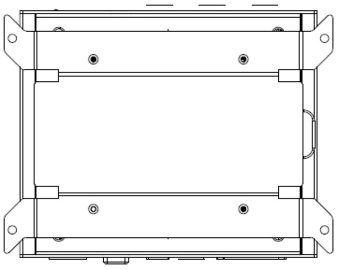


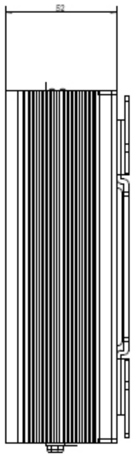
Mechanical dimensions of the LVC-2000 with wall-mounting kit

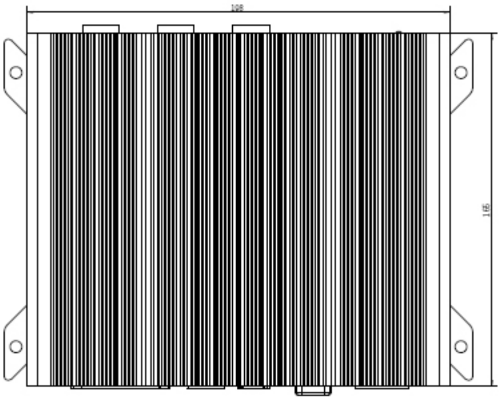
Unit: mm

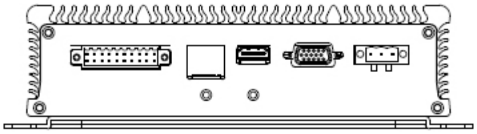






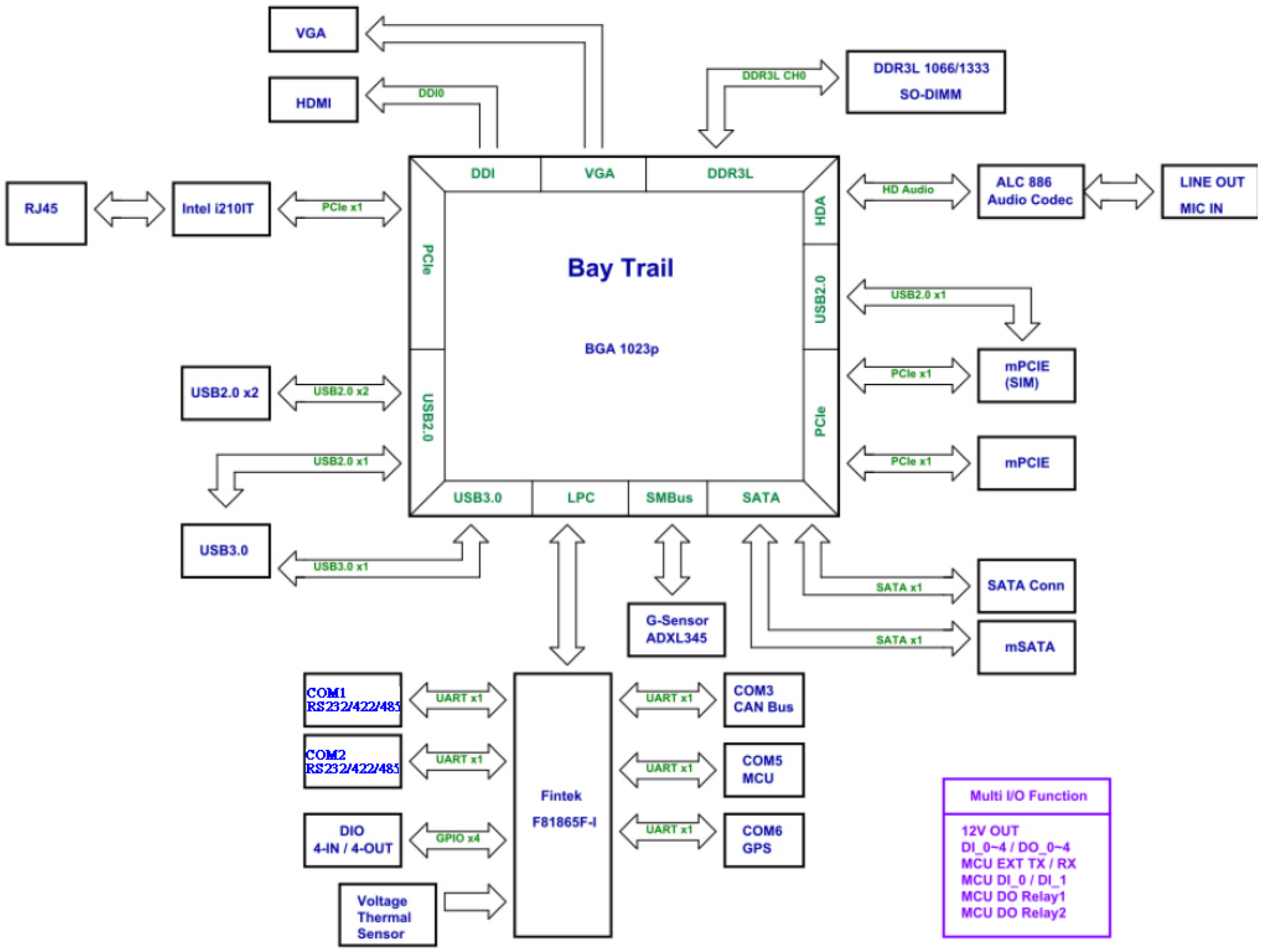




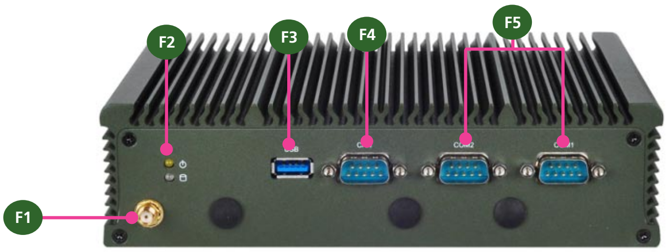


Block Diagram: The MainBoard

The block diagram depicts the relationships among the interfaces and modules on the motherboard.

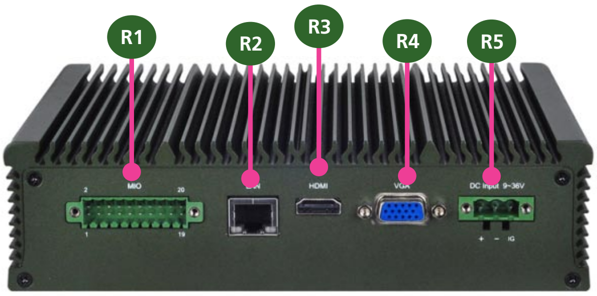


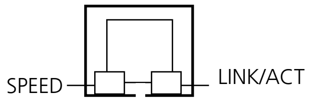
Front Components



Component	Description	Remarks
F1 GPS Antenna	Reserved for GPS antenna	
F2 HDD/SSD and Power LED (Green)	HDD/SSD • Blinking: means data access activities • Off: means no data access activities or no hard disk present Power • On: The computer is on. • Off: The computer is off .	
F3 USB 3.0 Ports	USB 3.0 type A connectors.	
F4 CAN bus	CAN bus connector for controller area network communication. It supports J1939 &J1708 standards.	Module required to enable it
F5 COM1/COM2	RS-232/422/485 ports for serial communication	

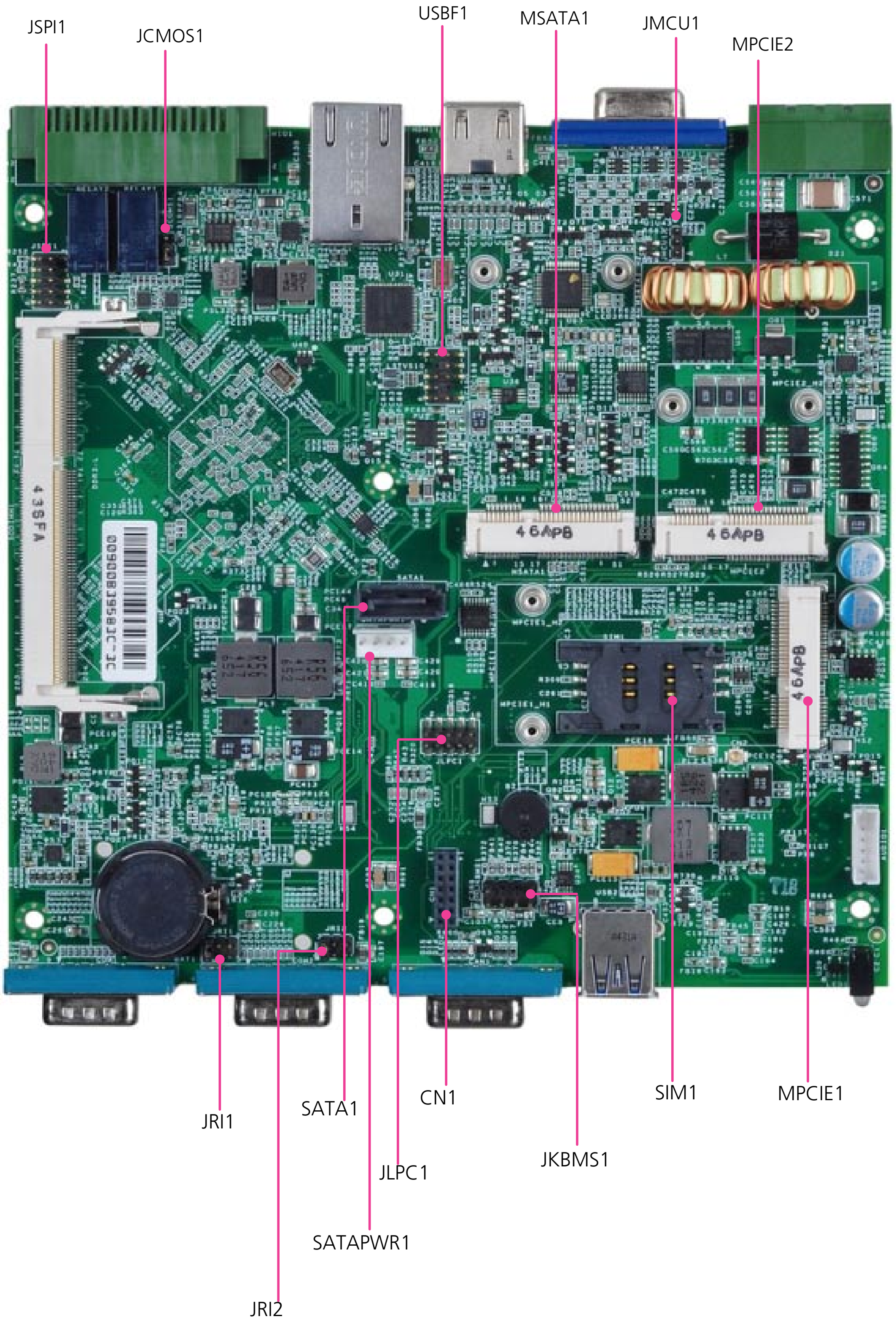
Rear Components



Component	Description	Remarks
R1 Multiple-I/O Connector	A 20-pin male connector for the following functions: • 4x DI (5V or 12V TTL selectable) • 4x DO (12V TTL , Max. 100mA) • 2x MCU DI • 2x Relay • 1x 12V Output @Max. 1A	
R2 One 10/100/1000Mbps LAN ports 	One RJ-45 (provided by Intel i210IT) jacks with LED indicators as described below LINK/ACT (Yellow) • On/Flashing: The port is linking and active in data transmission. • Off: The port is not linking. SPEED (Green/Amber) • Amber: The connection speed is 1000Mbps. • Green: The connection speed is 100Mbps • Off: The connection speed is 10Mbps.	
R3 HDMI Port (‡)	A HDMI port which is provided by Intel HD graphics (resolution: 1920x1080@60Hz).	
R4 VGA Port (‡)	It connects an external VGA monitor or projector (resolution: 1600x1200@60Hz)	
R5 Power-Input (DC)	Power-in with ignition support. The system support a wide range of power input +9~+36V including the prevalent 12V and 24V vehicular power system. It has a 2KV ESD protection on the DC input and ignition line.	

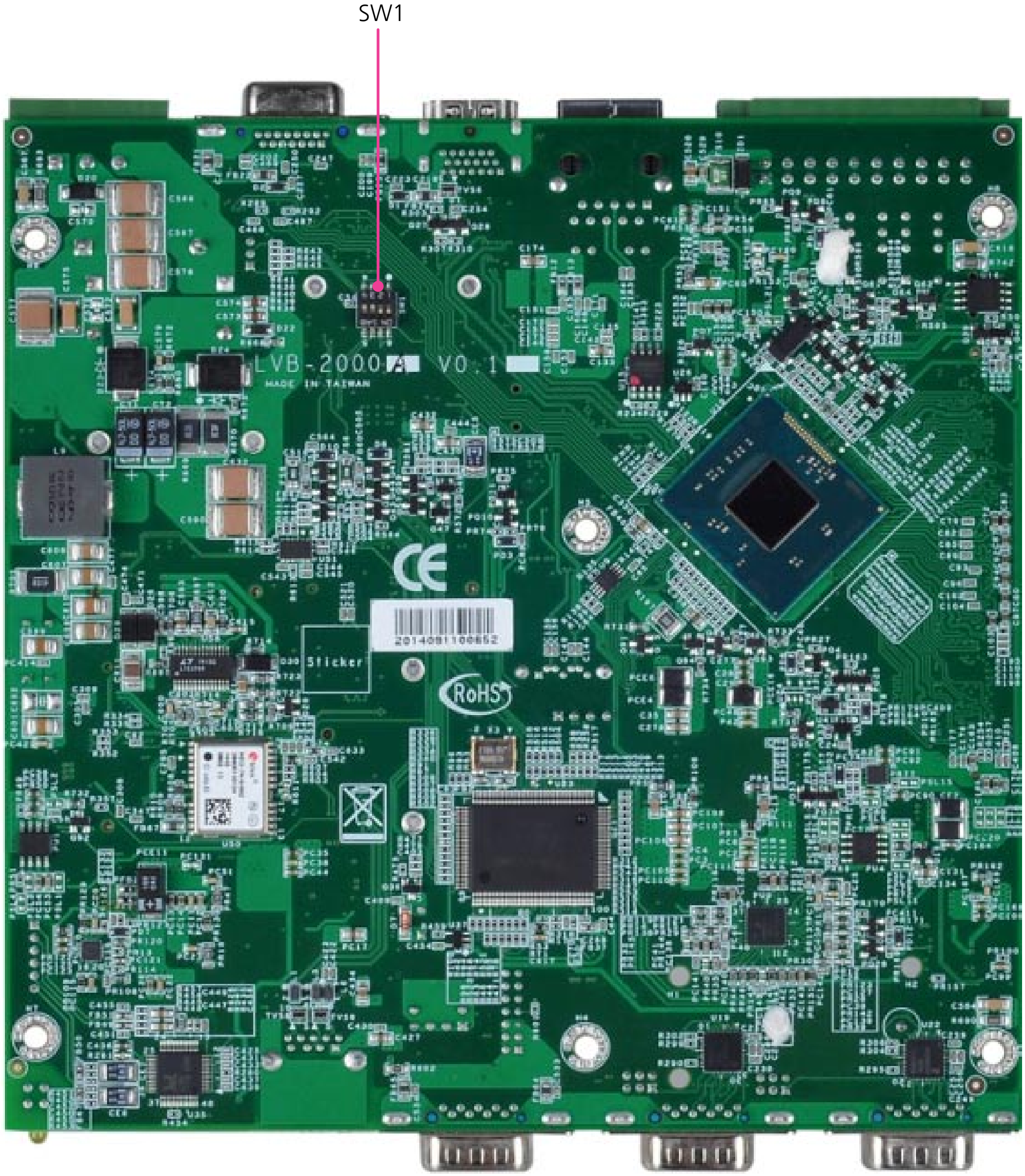
Internal Connectors and Jumpers

The following picture highlights the location of internal connectors and jumpers. Refer to the table 3.2 Connector List for more details.



Internal Connectors and Jumpers (backside)

The following picture highlights the location of internal connectors and jumpers on the backside of the board. Refer to the table 3.2 Connector List for more details.



Connectors and Jumpers List

The tables below list the function of each of the board jumpers and connectors by labels shown in the above section. The next section in this chapter gives pin definitions and instructions on setting jumpers.

Table 3.1 Connector List for External Connectors

Labels	**Function**
CAN1	CAN bus Connector
COM1/COM2	RS-232/422/485 Commmunication Ports
HDMI1	High Definition Multimedia Interface
MIO1	Multiple I/O Connectors
PRJK1	3-Pin DC-in Power Connector with Ignition Control
USB2	USB 3.0 Connector
VGA1	VGA Connector

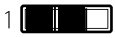
Table 3.2 Connector List for Internal Connectors

Labels	**Function**
AUDIO1	Audio Pin Header
JCMOS1	Clear CMOS Jumper
JMCU1	MCU Programming Jumper
JSPI1	Serial Peripheral Interface Bus
JLPC1	Low-pin Count Pin Header
JRI1/JRI2	COM1/COM2 Power Selection
MPCIE1/MPCIE2	Mini-PCIe Connector 1/2
mSATA1	mSATA Connector
JKBMS1	Keyboard/Mouse Connector
JRI1	COM1 Power Selection
JRI2	COM2 Power Selection
SATA1	SATA Driver Connector
SATAPWR1	SATA Power Connector
SIM1	SIM Card Connector
USBF1	USB 2.0 Pin Header

Jumper Settings

JCMOS1 (Clean CMOS):

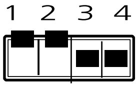
This jumper is used to erase data in CMOS. To clear CMOS, first turn off your system and unplug power source. Then, by placing the cap on pin 2 and 3 (short pin 2-3), this jumper can erase the system settings stored in CMOS memory.



Pin	Description
Short 1-2	Normal (default)
Short 2-3	Clear RTC

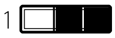
SW1: Function Select

Default Pin 1&2 ON; Pin 3&4 OFF



Pin	Status	Description
1	ON/OFF	Detect power good
2	ON/OFF	Low power detect
3	ON/OFF	Watchdog
4	ON/OFF	RSVD

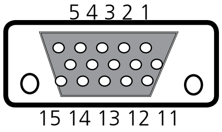
JMCU1: Programming the MCU



Pin	Description
Short 1-2	Program MCU
Short 2-3	Normal (default)

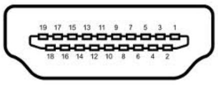
Connectors

VGA (VGA1)



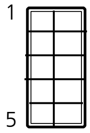
Pin	Signal	Pin	Signal	Pin	Signal
1	RED	6	GND	11	N/A
2	GREEN	7	GND	12	DDC DAT
3	BLUE	8	GND	13	VGA_HS
4	N/A	9	V5S_VGA	14	VGA_VS
5	GND	10	GND	15	VGA_CLK

HDMI (HDMI1)



Pin	Description	Pin	Description
1	HDMI_DATP2_P	2	GND
3	HDMI_DATP2_N	4	HDMI_DATP1_P
5	GND	6	HDMI_DATP1_N
7	HDMI_DATP0_P	8	GND
9	HDMI_DATP0_N	10	HDMI_CLK_P
11	GND	12	HDMI_CLK_N
13	N/A	14	N/A
15	HDMI_DDC_CLK	16	HDMI_DDC_DAT
17	GND	18	V5S_HDMI
19	HDMI_HPD		

USBF1 (USB0,1)

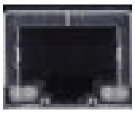


Pin	Description	Pin	Description
1	V5S_USB0	6	USBDN1
2	GND	7	USBDP0
3	N/A	8	N/A
4	USBDP1	9	GND
5	USBDN0	10	V5S_USB1

USB3.0 (USB2)

Pin	Description	Pin	Description
1	V5S_USB2	5	USB3_SSRXN
2	USBDN2	6	USB3_SSRXP
3	USBDP2	7	GND
4	GND	8	USB_SSTXP
		9	USB_SSTXN

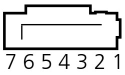
LAN (LAN1)



Pin	Description	Pin	Description
1	MDI_P0_LAN1	8	MDI_N2_LAN1
2	MDI_N0_LAN1	9	MDI_P3_LAN1
3	MDI_P1_LAN1	10	MDI_N3_LAN1
4	MDI_N1_LAN1	11	LNK100_LAN1
5	GND	12	LNK1000_LAN1
6	GND	13	V3P3A
7	MDI_P2_LAN1	14	ACT_LAN1

Serial-ATA Connector (SATA1):

SATA 7-pin signal connector for HDD/SSD. The interface signal is SATA 3.0 Gbps.

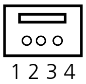


SATA1

Pin	Description
1	GND
2	SATATXP
3	SATATXN
4	GND
5	SATARXN
6	SATARXP
7	GND

4-pin Serial-ATA Power Connector (SATAPWR1): It is for connecting the SATA power cord.

SATAPWR1



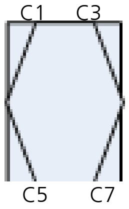
Pin	Description
1	VCC12
2	GND
3	GND
4	VCC5_PS

mSATA (MSATA1)

It complies with SATA 3.0Gbps interface

PIN	Description	PIN	Description
1	N/A	30	SMB_CLK
2	V3P3S	31	mSATATXN
3	N/A	32	SMB_DAT
4	GND	33	mSATATXP
5	N/A	34	GND
6	N/A	35	GND
7	N/A	36	N/A
8	N/A	37	GND
9	GND	38	N/A
10	N/A	39	V3P3S
11	N/A	40	GND
12	N/A	41	V3P3S
13	N/A	42	N/A
14	N/A	43	GND
15	GND	44	N/A
16	N/A	45	N/A
17	N/A	46	N/A
18	GND	47	N/A
19	N/A	48	N/A
20	N/A	49	N/A
21	GND	50	GND
22	N/A	51	N/A
23	mSATARXP	52	V3P3S
24	V3P3S	53	N/A
25	mSATARXN	54	N/A
26	GND	55	N/A
27	GND	56	N/A
28	N/A	57	N/A
29	GND	58	N/A

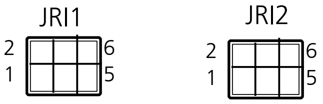
SIM card reader (SIM1)



Pin No.	Description
C1	UIM_PWR
C2	UIM_RST
C3	UIM_CLK
C5	GND
C6	UIM_VPP
C7	UIM_DAT

COM1/COM2 Power Selection (JRI1/JRI2):

JRI1 selects COM1 power voltage and JRI2 selects COM2 power voltage . The default is Ring Indicator (RI) for pin 8 of COM.



Pin No.	Signal
1-2	Default
3-4	VCC5
5-6	VCC12

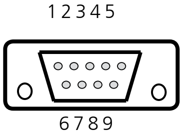
MPCIE1: Mini-PCIe Connector with one SIM Card Reader(SIM1). It supports both Wi-Fi and 3G module.

Pin	Signal	Pin	Signal
1	PCIE_WAKE_N	2	VCC3P3_PS
3	N/A	4	GND
5	N/A	6	V1P5_MPCIE
7	E_CLKREQ-	8	UIM_PWR
9	GND	10	UIM_DATA
11	PCIE_CKN3	12	UIM_CLK
13	PCIE_CKP3	14	UIM_RESET
15	GND	16	UIM_VPP
17	RSV	18	GND
19	RSV	20	N/A
21	GND	22	BUF_PLT_RST#
23	PCH_PCIE_RXN3	24	PCIE_PCIE_VCC3AUX
25	PCH_PCIE_RXP3	26	GND
27	GND	28	V1P5_MPCIE
29	GND	30	SMBCLK_RESUME
31	PCH_PCIE_TXN3	32	SMBDATA_RESUME
33	PCH_PCIE_TXP3	34	GND
35	GND	36	PCH_USB_N8
37	GND	38	PCH_USB_P8
39	VCC3P3_PS	40	GND
41	VCC3P3_PS	42	LED_WWAN1-
43	GND	44	LED_WLAN1-
45	RSV	46	N/A
47	RSV	48	V1P5_MPCIE
49	RSV	50	GND
51	RSV	52	VCC3P3_PS

MPCIE2: Mini-PCIe Connector (half-size)

Pin	Signal	Pin	Signal
1	PCIE_WAKE_N	2	VCC3P3_PS
3	N/A	4	GND
5	N/A	6	V1P5_MPCIE
7	E_CLKREQ-	8	UIM2_PWR
9	GND	10	RSV
11	PCIE_CKN4	12	RSV
13	PCIE_CKP4	14	RSV
15	GND	16	RSV
17	RSV	18	GND
19	RSV	20	N/A
21	GND	22	BUF_PLT_RST#
23	PCH_PCIE_RXN4	24	PCIE_PCIE_VCC3AUX
25	PCH_PCIE_RXP4	26	GND
27	GND	28	V1P5_MPCIE
29	GND	30	SMBCLK_RESUME
31	PCH_PCIE_TXN4	32	SMBDATA_RESUME
33	PCH_PCIE_TXP4	34	GND
35	GND	36	PCH_USB_N9
37	GND	38	PCH_USB_P9
39	VCC3P3_PS	40	GND
41	VCC3P3_PS	42	LED_WWAN2-
43	GND	44	LED_WLAN2-
45	RSV	46	N/A
47	RSV	48	V1P5_MPCIE
49	RSV	50	GND
51	RSV	52	VCC3P3_PS

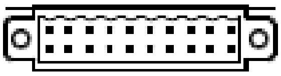
COM1&2



Pin	Description
1	SP1_DCD
2	SP1_DSR
3	SP1_RXD
4	SP1_RTS
5	SP1_TXD
6	SP1_CTS
7	SP1_DTR
8	VCC_RI1
9	GND

Pin	Description
1	SP2_DCD
2	SP2_DSR
3	SP2_RXD
4	SP2_RTS
5	SP2_TXD
6	SP2_CTS
7	SP2_DTR
8	VCC_RI2
9	GND

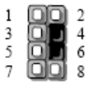
Multiple I/O Connectors (MIO1): Multiple I/O pins for functions in serial communication, Digital In/Out, Ignition detection input for automatic wake-up function



Pin No.	Function	Function
1	GND	
2	12V_OUT	**12VDC Power Output**
3	IGN_DI0	Input pin for automatic wakeup
4	IGN_DI1	Input pin for automatic wakeup
5	EXT_TXD_R	COM_TxD
6	EXT_RXD_R	COM_RxD
7	DI_0	Digital-In_0
8	DO_0	Digital-Out_0
9	DI_1	Digital-In_1
10	DO_1	Digital-Out_1
11	DI_2	Digital-In 2
12	DO_2	Digital-Out 2
13	DI_3	Digital-In 3
14	DO_3	Digital-Out 3
15	RELAY1_NOPEN	RELAY1 Normally Open
16	RELAY1_COMM	RELAY1 Common
17	GND	Ground
18	GND	Ground
19	RELAY2_NOPEN	RELAY2 Normally Open
20	RELAY2_COMM	RELAY2 Common

Maximum input/output current for each port is 100mA			
For all Input/ output pins:	Voltage	Logic	Register
	DI: <0.8V DO: <0.4V	Low	0
	DI: 10 ~ 12V DO:12V	High	1
The default BIOS value is 0 for DI and 1 for DO			
1. Pin3 and pin4 can be used for DI wake-up function (Refer to the flow chart in Chapter 4 and the ISM in Appendix A). 2. Pin 15, 16, 17 can be used for Digital output control with contact current 9~36V@2A (DO1); Pin 18, 19, 20 can be used for digital output control with contact current 9~36V@2A in maximum (DO2).			

Keyboard & Mouse connector (JKBMS1)

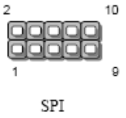


Pin No.	Pin Name	Pin No.	Pin Name
1	VCC5_KB	2	KCLK
3	MDATA		
5	KDATA		
7	GND	8	MCLK

Low-Pin Count (JLPC1)

Pin	Description	Pin	Description
1	33MHz CLK	2	LAD1
3	RESET	4	LAD0
5	FRAME	6	VCC 3.3
7	LAD3	8	GND
9	LAD2	10	GND

SPI (JSPI1)

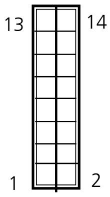


Pin	Description
1	SPI_HOLD
2	N/A
3	SPI_CS0
4	VCC3
5	SPI_MISO
6	N/A
7	N/A
8	SPI_CLK
9	GND
10	SPI_MOSI

Power Input with Ignition Control (PRJK1)

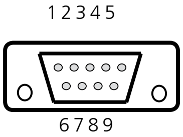
Pin No.	Pin Name
1	Ignition
2	GND
3	DC_IN

CAN Bus Module Connector (CN1)



Pin No.	Signal	Pin No.	Signal
1	BAT_12V_24V	2	K_LINE
3	DO	4	N/A
5	GND_CAN	6	GND_CAN
7	PLTRST_BUF1	8	J1850+/J1708+
9	SIO_SIN3	10	J1850-/J1708-
11	SIO_SOUT3	12	CAN_H/J1939+
13	V5S	14	CAN_L/J1939-

CAN Bus Connector (CAN1)



Pin No.	Pin Name
1	J1850-/J1708-
2	GND_CAN
3	CAN_H/J1939+
4	K_LINE
5	CAN_L/J1939-
6	J1850-/J1708-
7	J1850+/J1708+
8	J1850+/J1708+
9	BAT_12V_24V

AUDIOIN1: Line-out and Mic-in Connector

Pin No.	Pin Name
1	MIC_IN_L
2	MIC_IN_R
3	GND_AUO
4	GND_AUO
5	FRONT_OUT_L
6	FRONT_OUT_R

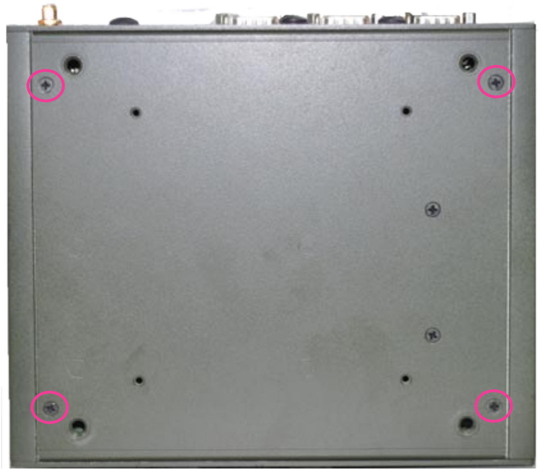
Chapter 3: Hardware Setup

Preparing the Hardware Installation

To access some components and perform certain setup, please read the warning below before installation procedures.

WARNING: To reduce the risk of personal injury, electric shock, or damage to the equipment, remove the power cord to remove power from the server. The power switch button does not completely shut off system power. Portions of the power supply and some internal circuitry might remain active until power is removed.

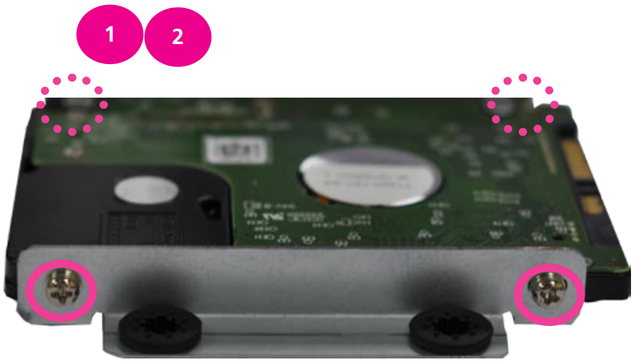
1. Unpower the LVC-2000 and remove the power cord.
2. Remove 4 threaded screws from the bottom to take off the bottom cover.
3. Open the cover.

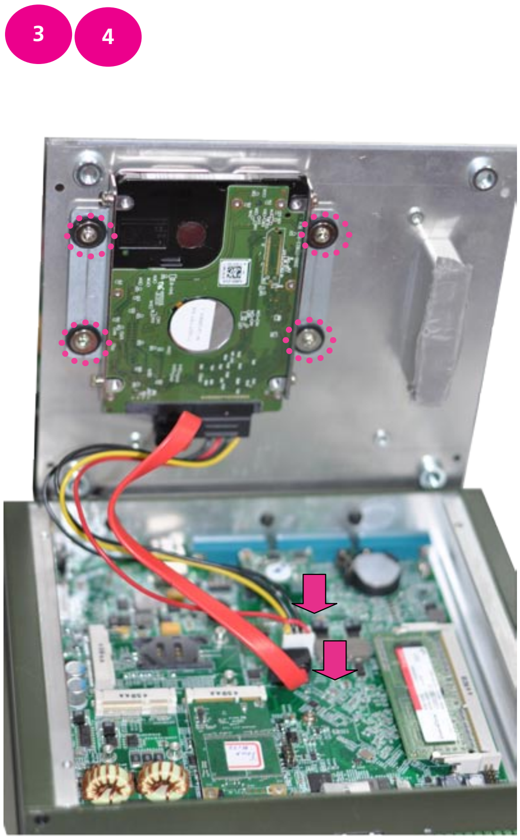


Disk Drive Installation

The system can accommodate one Serial-ATA 2.5" HDD/ SSD. Follow these steps to install a disk drive into the system:

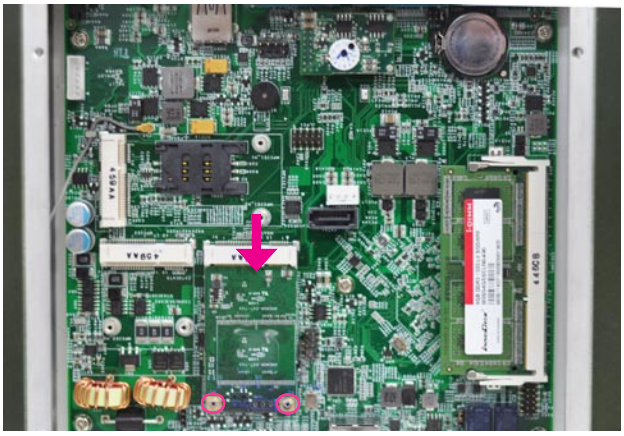
1. Take out the hard disk tray and place the disk drive on the tray with 4 mounting screws as illustrated in the following picture.
2. Plug the Serial-ATA cable to the hard disk.
3. Attach the disk drive to the system's chassis and secure it with the mounting screws.
4. Connect the Serial-ATA power and data disk cables to the Serial-ATA power and disk connectors on the main board respectively.





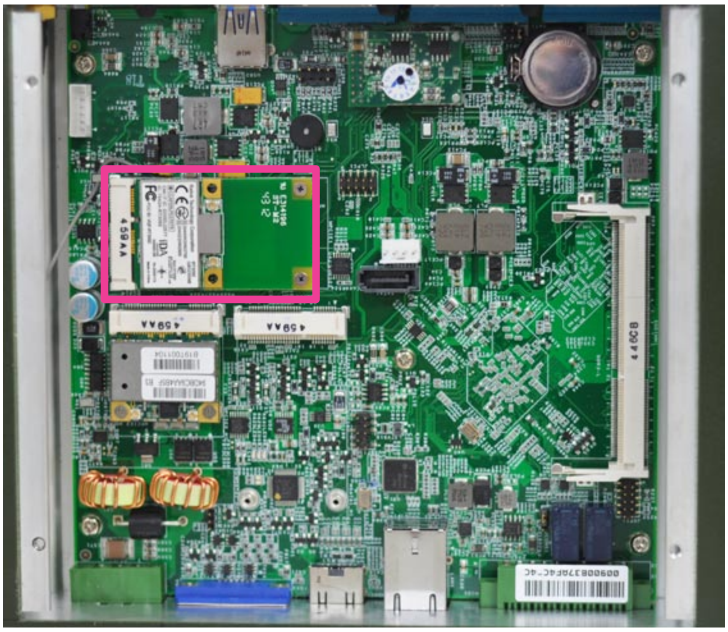
mSATA Card Installation

1. Align the mSATA card's key with the Mini-PCIe slot notch.
2. Insert the wireless module into the connector diagonally.
3. Install the module onto the board with the screws.



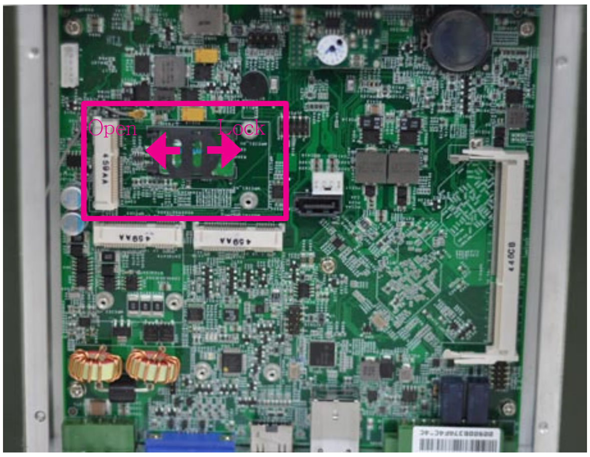
Wireless Module Installation

1. Align the wireless module's cutout with the Mini-PCIe slot notch.
2. Insert the wireless module into the connector diagonally.
3. Push the other end of the wireless module to be tightened with the latch. Then, install the module with screws



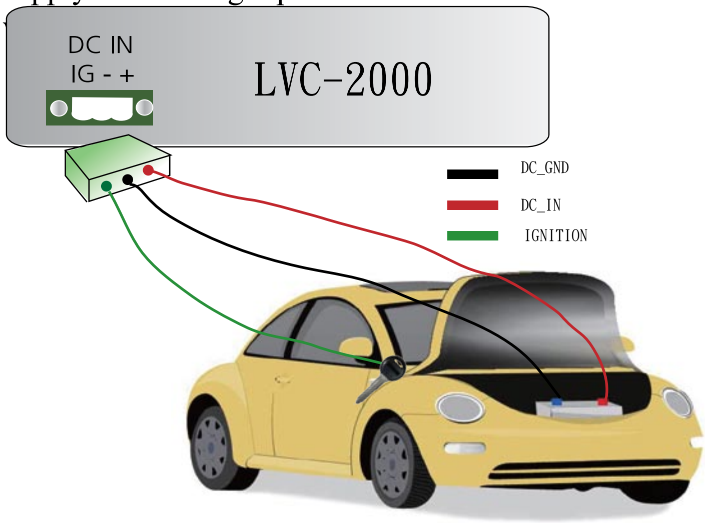
3G SIM Card Installation

1. Unlock the SIM card reader.
2. Place the SIM card on the SIM card reader. Notice the angled corner to align the SIM card properly.
3. Lock the SIM card reader.



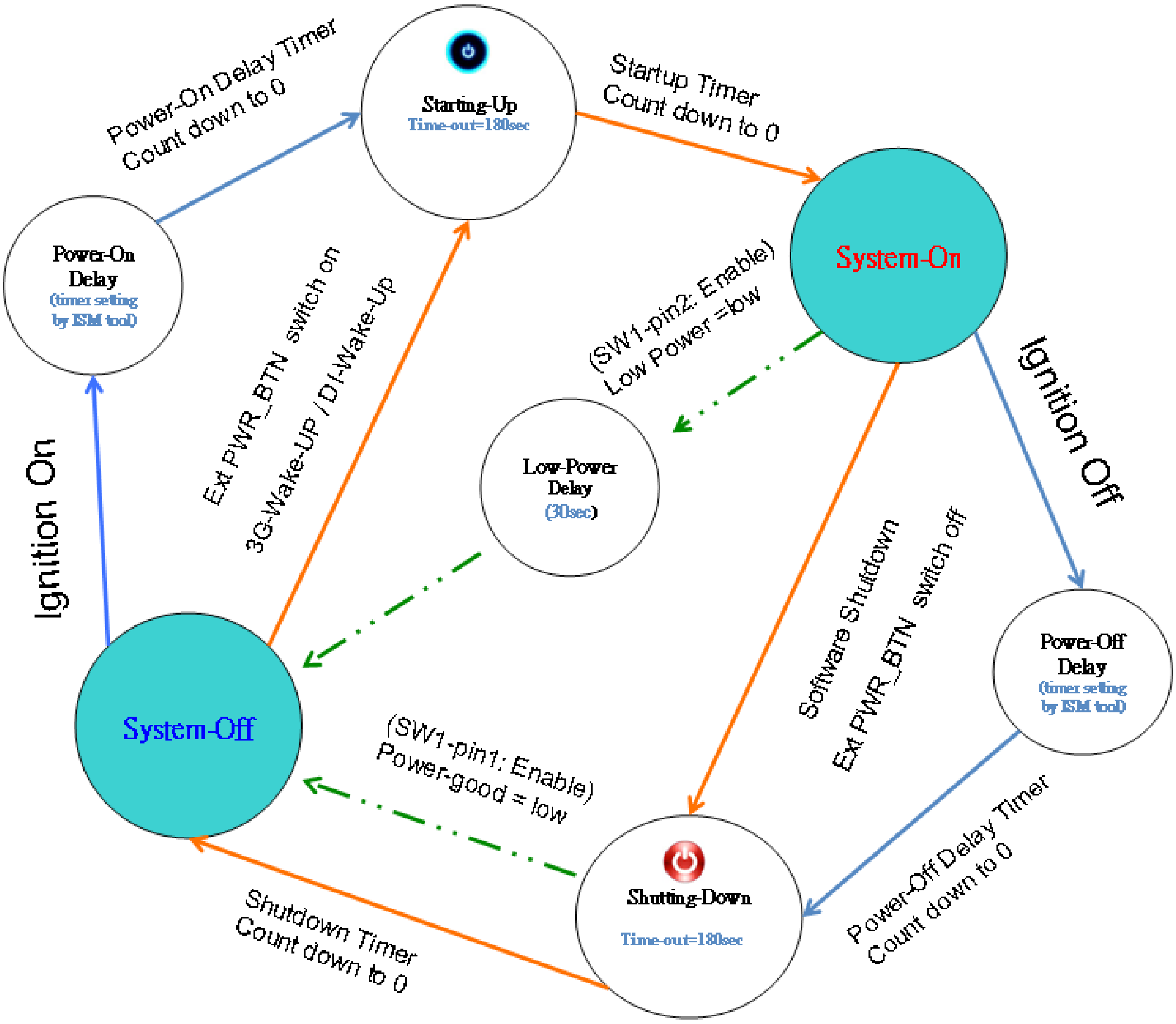
Connecting Power

Connect the LVC-2000 to a +9V ~ +36V vehicle battery. The DC power-in connector comes with a 3-pin terminal block for its Phoenix contact. This power socket can only accept the power supply with the right pin contact so be cautious



Chapter 4:
The Flow Chart

The flow chart section contains all flow chart used in the system. The flow chart describes the system's behavior on powering on and off the system via power ignition control or on/off switch when the appropriate timer control parameters are set.



Note:

1. For power-good and low-voltage mechanism to function in the workflow, you will need to enable the power-good and low-voltage detection function with ***selector 1*** and ***selector 2*** jumper respectively of ***SW1***. (Refer to *Chapter 3 Board Layout*).
2. For power on and power off delay timer parameter, refer to *Appendix A Using the Ignition System Manager (ISM)*.
3. For DI wake-up function, refer to jumper MIO2 ***Pin NO.19*** and **21**. Refer to *Chapter 3 Board Layout* and *Appendix A Using the Ignition System Manager (ISM)* for jumper setting and parameter setting respectively.
4. When the system's shutdown timer start counting down 180sec, using ignition or External PWR_BTN to start the system again during shutdown process will not work until the countdown finishes.

Appendix A: Using the Ignition System Manager (ISM)

The Ignition System Manager (ISM) is a software that can monitor the system's voltage level and configure the features that the Power Ignition Module provides.

For sample ISM code, see *ISM* folder under LVC-2000 Utility on the *Driver and Manual CD.*

Running the Program

Just double click the ISM.exe to launch the ISM.

The program can configure the following values:

Voltage: It shows the current power system.

Power Input System: Select either 12V or 24V for vehicular power input.

Startup Voltage (V): If the DC-in voltage is not higher than this value, the system will not be able to start up.

Shutdown Voltage (V): If the DC-in voltage is lower than the shutdown voltage, the system will start shutdown process automatically. (Refer to selector 2 of SW1 dip switch on the mainboard.)

Power-on Delay (min/sec): Select power-on delay value to indicate the time to delay powering on the system. (Refer to the flow chart in Chapter 4)

Power-off Delay (hr/min/sec): Select power-off delay value to indicate the time to delay powering off the system (Refer to the flow chart in Chapter 4)

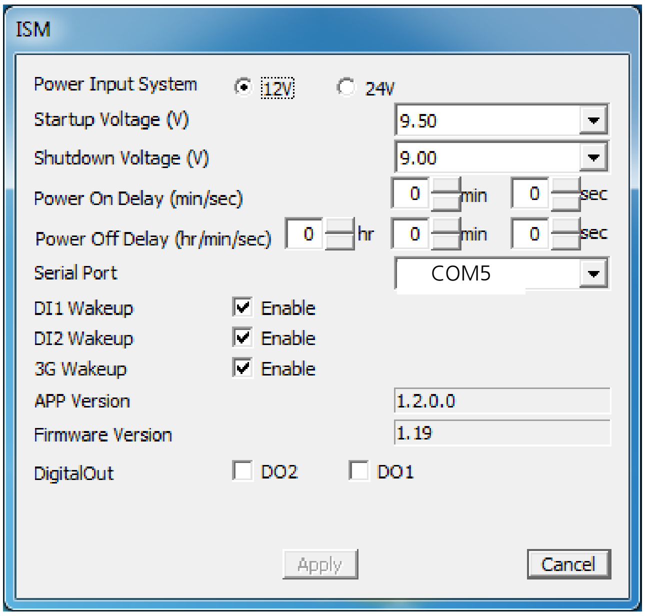
Serial Port: Select the serial communication port for the ISM. Choose COM5.

D1/D2 Wakeup: Digital input triggering to enable automatic wake-up function. Select this option and it will start the system automatically once an input has been triggered.

3G Wakeup: 3G SMS/Ring wake-up to enable automatic wake-up function. Select this option and it will start the system automatically through 3G Internet service.

DigitalOut: Check the box to turn on the output device and check off the box to turn off the connected device.

After you have made changes, click **Apply** to apply the changes to the Ignition controller or **Cancel** to cancel the changes.



ISM

Power Input System ● 12V ○ 24V

Startup Voltage (V) 9.50

Shutdown Voltage (V) 9.00

Power On Delay (min/sec) 0 min 0 sec

Power Off Delay (hr/min/sec) 0 hr 0 min 0 sec

Serial Port COM5

DI1 Wakeup ☑ Enable

DI2 Wakeup ☑ Enable

3G Wakeup ☑ Enable

APP Version 1.2.0.0

Firmware Version 1.19

DigitalOut □ DO2 □ DO1

Note:

1. You will have to enable (the default is enabled) the *selector 2 (Low Voltage Detection) of SW1 dip switch* on the mainboard to enable automatic shutdown function. (Refer to *Select MCU Detect Function for power ignition behavior (SW1)* in **Chapter 3 Board Layout**.)
2. DI1/DI2 Wakeup function is detected via pin 19/21of MIO2 (Refer to *MIO2* in **Chapter 3 Board Layout**.)
3. DO1 function is connected (controlled) via pin 20, 22, 23 while DO2 is connected (controlled) via pin 24, 25, 26. (Refer to *MIO2* in **Chapter 3 Board Layout**.)
4. Refer to the flow charts in Chapter 4 for more information.

Appendix B: Digital Input/Output

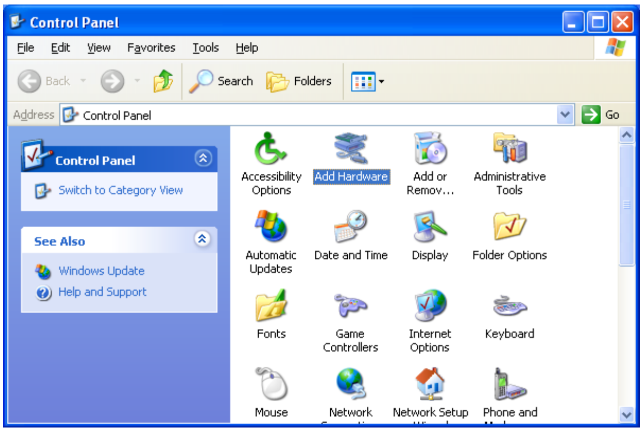
The Digitanl I/O on the rear panel is designed to provide the input and output operations for the system. For sample DIO code, see SuperIO folder under LVC-2000 Utility on the *Driver and Manual CD*. Make sure that you have installed the Lanner GPIO driver as instructed below.

Driver Installation

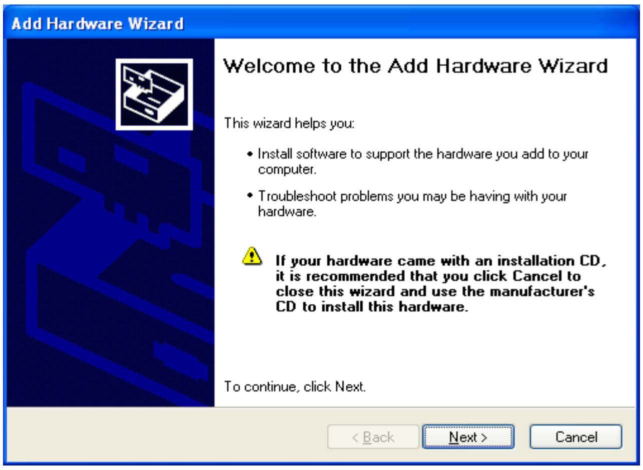
Before you could access or control the operation of the G-sensor, GPS and Digital I/O functions, install the the L_IO driver which is the library and driver needed for Lanner General Purpose Input/Output interface or functions.

To install the L_IO driver:

1. Restart the computer, and then log on with Administrator privileges.
2. Insert the Drivers and User's Manual CD to the USB-optical drive.
3. Browse the contents of the support CD to locate the file in the LIO folder.
4. From the control panel, click the ADD Hardware program



5. Select Next to proceed



Add Hardware Wizard

Welcome to the Add Hardware Wizard

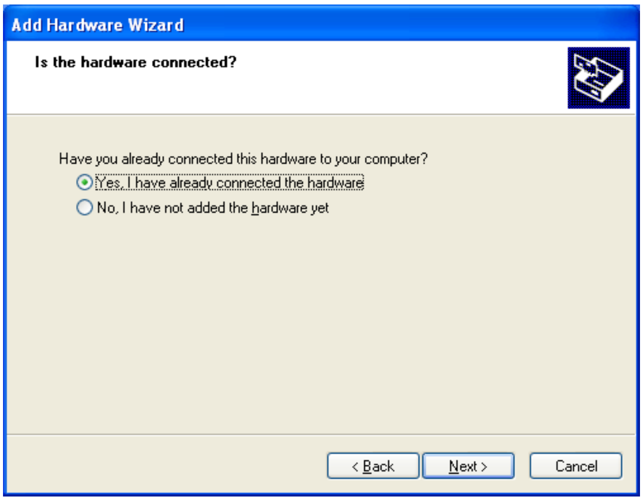
This wizard helps you:

- Install software to support the hardware you add to your computer.
- Troubleshoot problems you may be having with your hardware.

If your hardware came with an installation CD, it is recommended that you click Cancel to close this wizard and use the manufacturer's CD to install this hardware.

To continue, click Next.

6. Answer "Yes" to the question and select Next to proceed.



Add Hardware Wizard

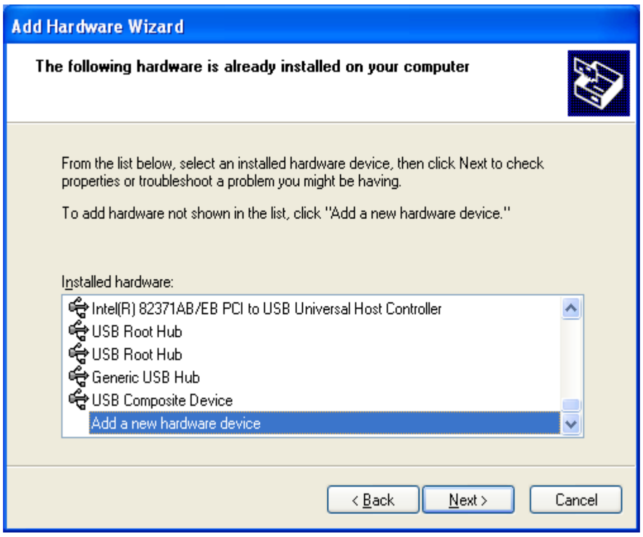
Is the hardware connected?

Have you already connected this hardware to your computer?

● Yes, I have already connected the hardware

○ No, I have not added the hardware yet

7. Select Add a new hardware device.



Add Hardware Wizard

The following hardware is already installed on your computer

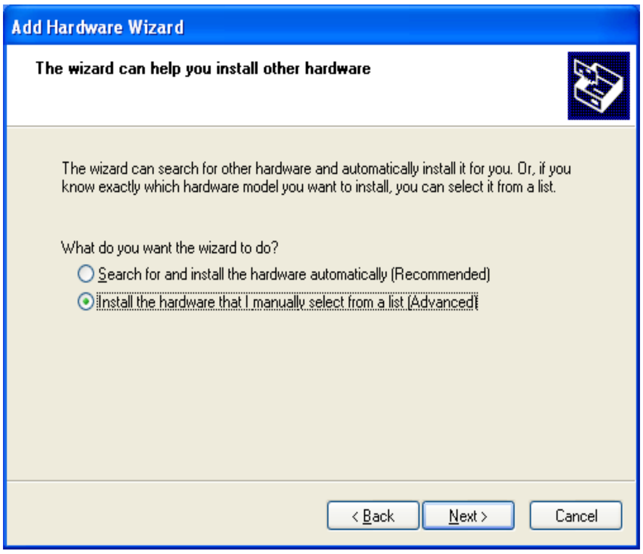
From the list below, select an installed hardware device, then click Next to check properties or troubleshoot a problem you might be having.

To add hardware not shown in the list, click "Add a new hardware device."

Installed hardware:

- Intel(R) 82371AB/EB PCI to USB Universal Host Controller
- USB Root Hub
- USB Root Hub
- Generic USB Hub
- USB Composite Device
- Add a new hardware device

1. Choose to select the hardware Manually



Add Hardware Wizard

The wizard can help you install other hardware

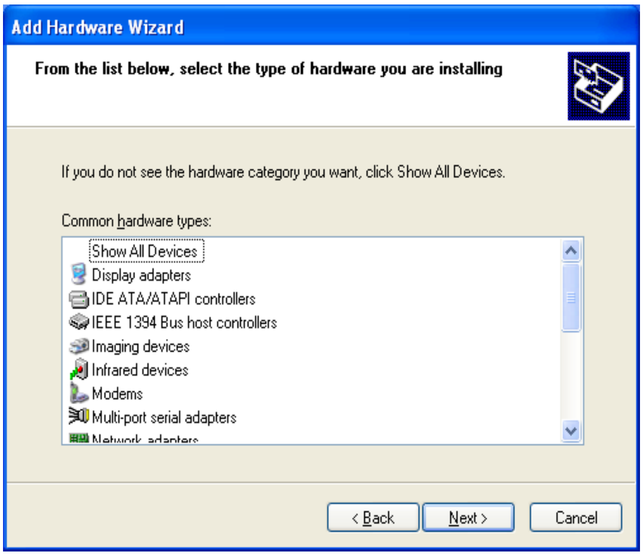
The wizard can search for other hardware and automatically install it for you. Or, if you know exactly which hardware model you want to install, you can select it from a list.

What do you want the wizard to do?

○ Search for and install the hardware automatically (Recommended)

● Install the hardware that I manually select from a list (Advanced)

2. Choose Show all device and click Next.



Add Hardware Wizard

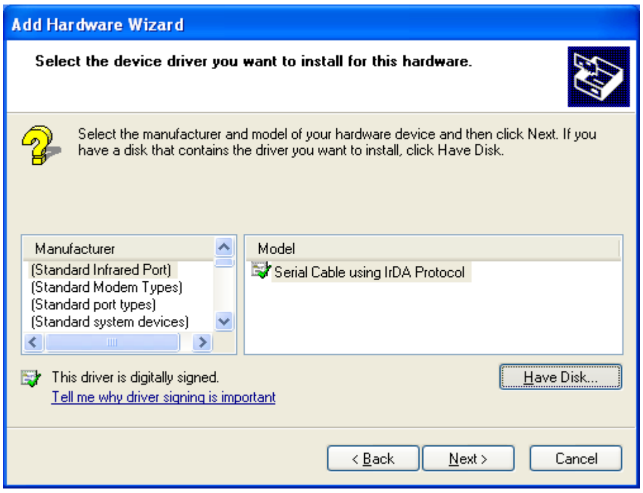
From the list below, select the type of hardware you are installing

If you do not see the hardware category you want, click Show All Devices.

Common hardware types:

- Show All Devices
- Display adapters
- IDE ATA/ATAPI controllers
- IEEE 1394 Bus host controllers
- Imaging devices
- Infrared devices
- Modems
- Multi-port serial adapters
- Network adapters

3. Click HaveDisk to locate the L_IO.inf file



Add Hardware Wizard

Select the device driver you want to install for this hardware.

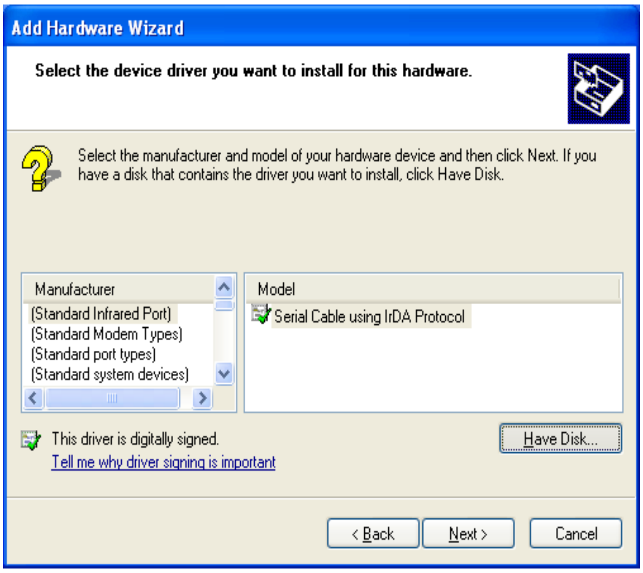
Select the manufacturer and model of your hardware device and then click Next. If you have a disk that contains the driver you want to install, click Have Disk.

Manufacturer	Model
(Standard Infrared Port)	Serial Cable using IrDA Protocol
(Standard Modem Types)	
(Standard port types)	
(Standard system devices)	

This driver is digitally signed.

Tell me why driver signing is important

4. Click HaveDisk to locate the L_IO.inf file



Add Hardware Wizard

Select the device driver you want to install for this hardware.

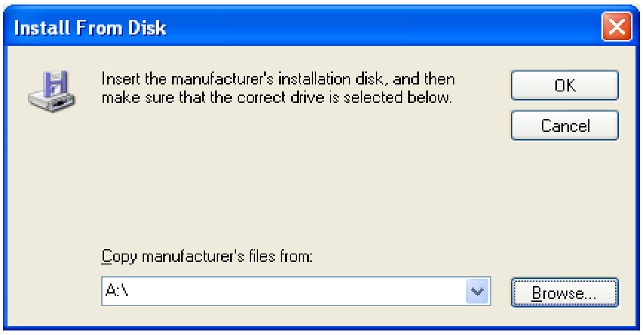
Select the manufacturer and model of your hardware device and then click Next. If you have a disk that contains the driver you want to install, click Have Disk.

Manufacturer	Model
(Standard Infrared Port)	Serial Cable using IrDA Protocol
(Standard Modem Types)	
(Standard port types)	
(Standard system devices)	

This driver is digitally signed.

Tell me why driver signing is important

5. Select the L_IO.inf



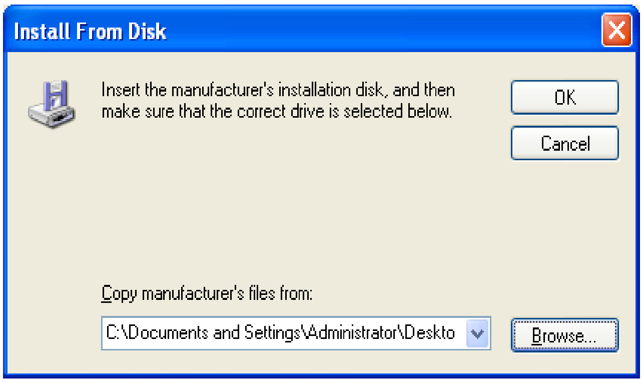
Install From Disk

Insert the manufacturer's installation disk, and then make sure that the correct drive is selected below.

Copy manufacturer's files from:

A:\

6. Select OK to confirm with the installation



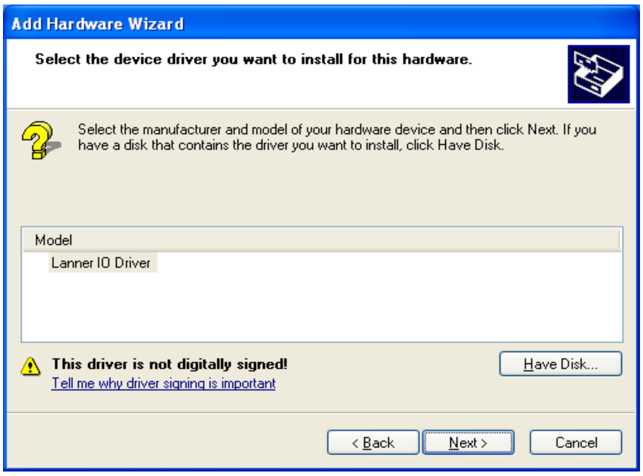
Install From Disk

Insert the manufacturer's installation disk, and then make sure that the correct drive is selected below.

Copy manufacturer's files from:

C:\Documents and Settings\Administrator\Deskto

1. Select the Lanner IO driver and click Next.


Add Hardware Wizard

Select the device driver you want to install for this hardware.

Select the manufacturer and model of your hardware device and then click Next. If you have a disk that contains the driver you want to install, click Have Disk.

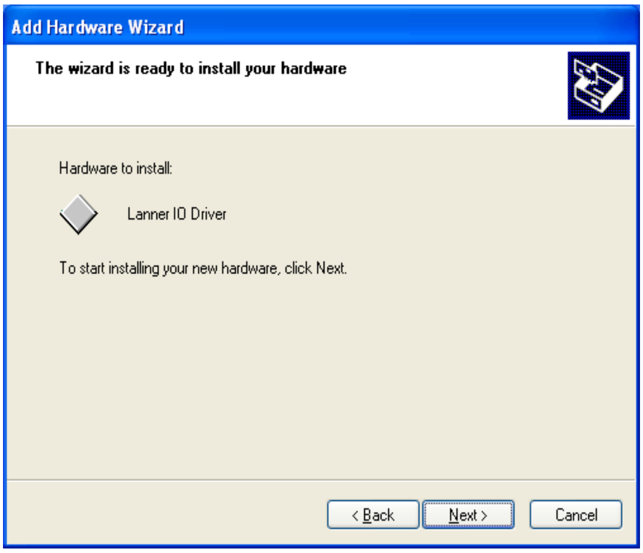
Model

Lanner IO Driver

This driver is not digitally signed!

Tell me why driver signing is important

2. Click Next


Add Hardware Wizard

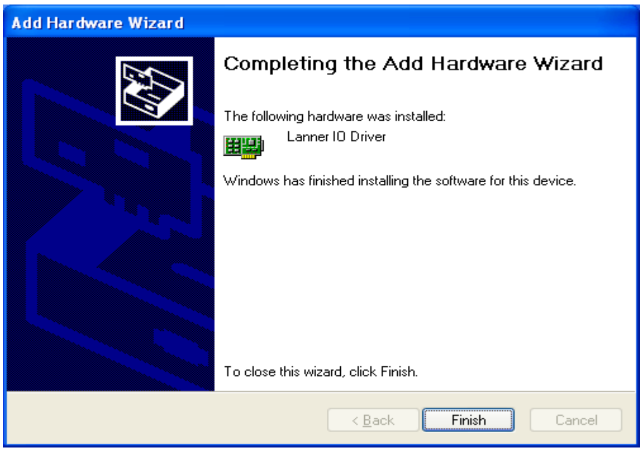
The wizard is ready to install your hardware

Hardware to install:

Lanner IO Driver

To start installing your new hardware, click Next.

3. Click **Complete** to close the installation program.


Add Hardware Wizard

Completing the Add Hardware Wizard

The following hardware was installed:

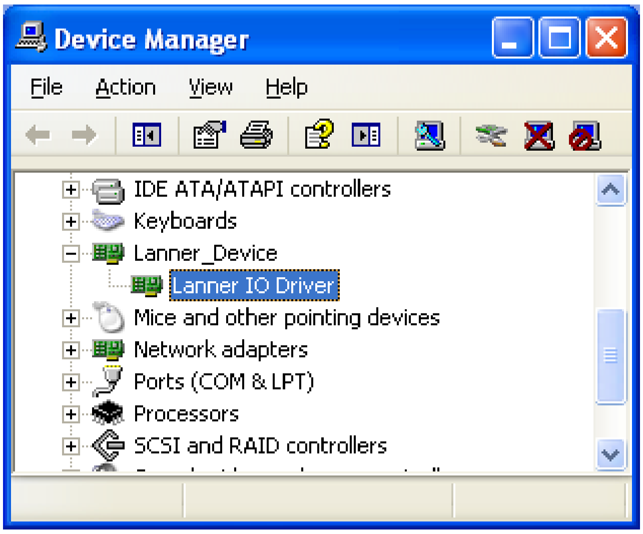
Lanner IO Driver

Windows has finished installing the software for this device.

To close this wizard, click Finish.

To verify the GPIO driver installation, do the following steps:

1. Right-click on the My Computer icon, and then select Properties form the menu.
2. Click the Hardware tab, then click the Device Manager button.
3. Click the + sign next to the Lanner_Device, then the Lanner IO Driver should be listed.


Device Manager

- IDE ATA/ATAPI controllers
- Keyboards
- Lanner_Device
  - Lanner IO Driver
- Mice and other pointing devices
- Network adapters
- Ports (COM & LPT)
- Processors
- SCSI and RAID controllers

A sample DIO program in C:

```
 ioaccess.c: IO access code for Lanner Platfomr Digital IO program
**********************************************************
**********************/



#include “../include/config.h”


#ifdef DJGPP

/* standard include file */
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
/* For DOS DJGPP */
#include <dos.h>
#include <inlines/pc.h>

#else //DJGPP
/* For Linux */


#ifdef DIRECT_IO_ACCESS
/* For Linux direct io access code */
/* standard include file */
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

#if defined(LINUX_ENV)
#include <sys/io.h>
#endif

#if defined(FreeBSD_ENV)
#include <machine/cpufunc.h>
#endif
```

```
#include <time.h>
#include <stdint.h>
#include <fcntl.h>
#include <errno.h>
#include <string.h>
#define delay(x) usleep(x)
#endif


#ifdef MODULE

#include <linux/kernel.h>
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/fs.h>
#include <asm/io.h>
#include <linux/delay.h>

#undef delay
#define delay(x) mdelay(x)
#undef fprintf
#define fprintf(S, A)  printk(A)

#endif //MODULE


#ifdef KLD_MODULE

#include <sys/types.h>
#include <sys/param.h>
#include <sys/systm.h>
#include <sys/malloc.h>
#include <sys/kernel.h>
#include <sys/bus.h>
#include <sys/errno.h>
```

```
#include <machine/bus.h>
#include <machine/resource.h>

#endif


#endif


/* local include file */
#include "../include/ioaccess.h"


#if (defined(MODULE) || defined(DIRECT_IO_ACCESS) ||
defined(KLD_MODULE))


/*
 *-----------------------------------------------------------------------------
-----
 * LEB-5000 Version V1.0
 *output3-0 = GPIO 03-00, input3-0= GPIO 53-50
 *-------------------------------------------------------------------------------
---------
 */


/*
 * Device Depend Definition :
*/


#define INDEX_PORT       0x2E
#define DATA_PORT        0x2F

void enter_SIO_config(void)
{
    outportb(INDEX_PORT, 0x87); // Must Do It Twice
    outportb(INDEX_PORT, 0x87);
    return;
}



void exit_SIO_config(void)
{
    outportb(INDEX_PORT, 0xAA);
    return;
}



unsigned char read_SIO_reg(int LDN, int reg)
{
        outportb(INDEX_PORT, 0x07); //LDN register
    delay(5);
    outportb(DATA_PORT, LDN);
    delay(5);
    outportb(INDEX_PORT, reg);
    delay(5);
    return(inportb(DATA_PORT));
}



void write_SIO_reg(int LDN, int reg, int value)
{


    outportb(INDEX_PORT, 0x07); //LDN register
    delay(5);
    outportb(DATA_PORT, LDN);
    delay(5);
    outportb(INDEX_PORT, reg);
    delay(5);
    outportb(DATA_PORT, value);
    return;
}


void dio_gpio_init(void)
{
        enter_SIO_config();
        write_SIO_reg(0x6, 0x30,0x01);      //enable GPIO
Port
        write_SIO_reg(0x6, 0xf0,((read_SIO_reg(0x6,
0xf0)& 0xF0)|0x0f)); //RxF0[3-0]=1111b, output
        write_SIO_reg(0x6, 0xA0, (read_SIO_reg(0x6,
0xA0)& 0xF0));  //RxA0[3-0]=0000b,          input
```

```
              exit_SIO_config();
       return;
}
void dio_set_output(unsigned char out_value)
{
              enter_SIO_config();
              write_SIO_reg(0x6, 0xf1, ((read_SIO_reg(0x6,
0xf1)& 0xF0)|out_value));
              exit_SIO_config();
       return;
}

unsigned int dio_get_input(void)
{
       unsigned int tmp=0x00;
              enter_SIO_config();
              tmp=read_SIO_reg(0x6, 0xA2)& 0x0f;
              exit_SIO_config();
        return tmp;
}


//======================================
=========================================
================
#endif
```

Appendix C: Accessing the GPS Data from the LVC-2000

The LVC-2000 employs an onbard u-blox NEO-7N GPS module for vehicle tracking and navigation system. You could read the GPS data through the RS-232 serial port.

It has the following listed key features and performance ratings:

Receiver type	50 Channels GPS L1 frequency, C/A Code SBAS: WAAS, EGNOS, MSAS
Time-To-First-Fix (All satellites at -130 dBm)	Cold Start: 26 s Warm Start: 26 s Hot Start: 1 s Aided Starts: 1 s
Sensitivity	• Tracking &Navigation: -162dBm • Reacquisition: -160dBm • Cold Start (without aiding): -148 dBm • Hot Start: -157 dBm
Maximum Navigation update rate	5Hz
Horizontal position accurach (CEP, 50%, 24 hours static, -130dBm, SEP: <3.5m)	GPS: 2.5m SBAS: 2.0m
Configurable Timepulse frequency range	0.25 Hz to 1 kHz
Accuracy for Timepulse signal	RMS: 30 ns 99%: <60 ns Granularity: 21 ns Compensated: 15 ns
Velocity accuracy	0.1m/s
Heading accuracy	0.5 degrees

Receiver type	50 Channels GPS L1 frequency, C/A Code SBAS: WAAS, EGNOS, MSAS
Time-To-First-Fix (All satellites at -130 dBm)	Cold Start: 26 s Warm Start: 26 s Hot Start: 1 s Aided Starts: 1 s
Operational Limits	Dynamics: less than and equal to 4g Altitude: 50,000m Velocity: 500m/s (Assuming Airborne <4g platform)

Specify the following communication parameters:

Bits per Second: 9600

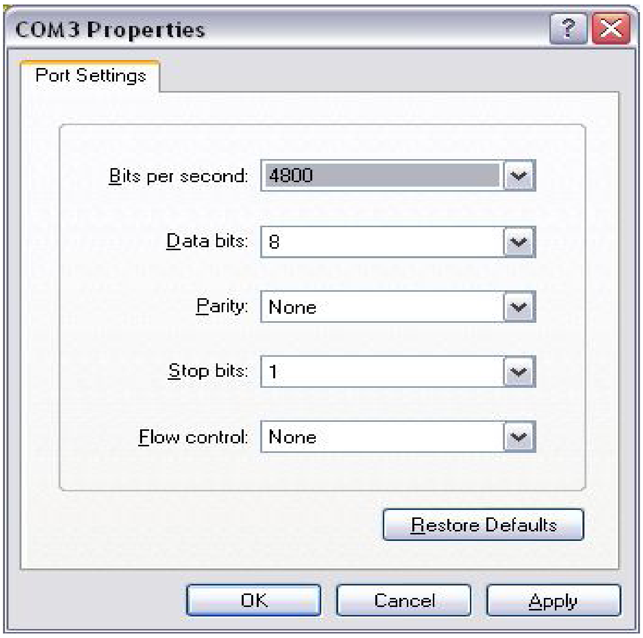
Data Bits: 8

Parity: None

Stop Bit: 1

Flow Control: None

9600



COM3 Properties

Port Settings

Bits per second: 4800

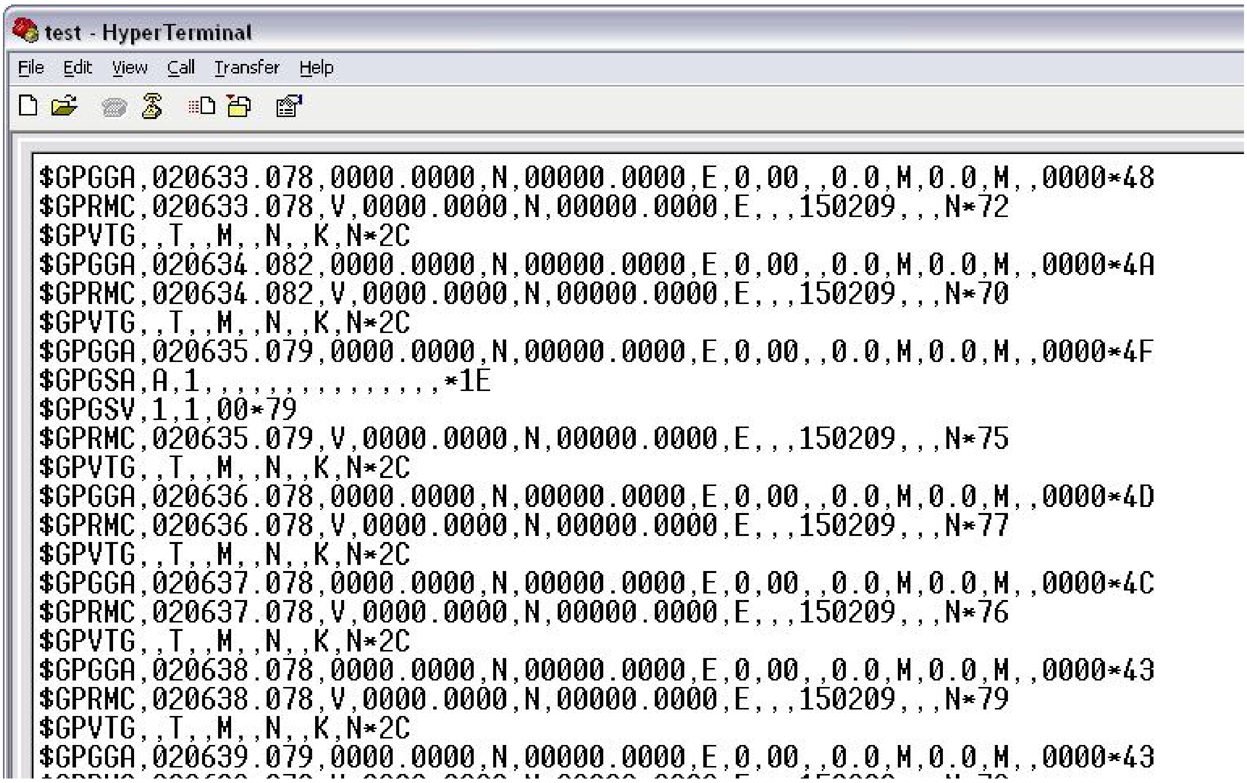
Data bits: 8

Parity: None

Stop bits: 1

Flow control: None

The hyper terminal should display GPS data:



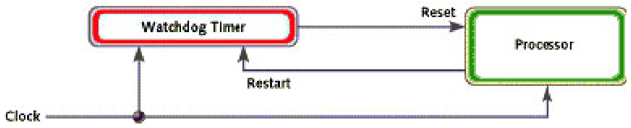
test - HyperTerminal

```
$GPGGA,020633.078,0000.0000,N,00000.0000,E,0,00,,0.0,M,0.0,M,,0000*48
$GPRMC,020633.078,V,0000.0000,N,00000.0000,E,,,150209,,,N*72
$GPVTG,,T,,M,,N,,K,N*2C
$GPGGA,020634.082,0000.0000,N,00000.0000,E,0,00,,0.0,M,0.0,M,,0000*4A
$GPRMC,020634.082,V,0000.0000,N,00000.0000,E,,,150209,,,N*70
$GPVTG,,T,,M,,N,,K,N*2C
$GPGGA,020635.079,0000.0000,N,00000.0000,E,0,00,,0.0,M,0.0,M,,0000*4F
$GPGSA,A,1,,,,,,,,,,,,,,,*1E
$GPGSV,1,1,00*79
$GPRMC,020635.079,V,0000.0000,N,00000.0000,E,,,150209,,,N*75
$GPVTG,,T,,M,,N,,K,N*2C
$GPGGA,020636.078,0000.0000,N,00000.0000,E,0,00,,0.0,M,0.0,M,,0000*4D
$GPRMC,020636.078,V,0000.0000,N,00000.0000,E,,,150209,,,N*77
$GPVTG,,T,,M,,N,,K,N*2C
$GPGGA,020637.078,0000.0000,N,00000.0000,E,0,00,,0.0,M,0.0,M,,0000*4C
$GPRMC,020637.078,V,0000.0000,N,00000.0000,E,,,150209,,,N*76
$GPVTG,,T,,M,,N,,K,N*2C
$GPGGA,020638.078,0000.0000,N,00000.0000,E,0,00,,0.0,M,0.0,M,,0000*43
$GPRMC,020638.078,V,0000.0000,N,00000.0000,E,,,150209,,,N*79
$GPVTG,,T,,M,,N,,K,N*2C
$GPGGA,020639.079,0000.0000,N,00000.0000,E,0,00,,0.0,M,0.0,M,,0000*43
```

Appendix D: Programming System Watchdog Timer of the LVC-2000

A watchdog timer is a piece of hardware that can be used to automatically detect system anomalies and reset the processor in case there are any problems. Generally speaking, a watchdog timer is based on a counter that counts down from an initial value to zero. The software selects the counter's initial value and periodically restarts it. Should the counter reach zero before the software restarts it, the software is presumed to be malfunctioning and the processor's reset signal is asserted. Thus, the processor will be restarted as if a human operator had cycled the power.

For sample watchdog code, see *watchdog* folder under LVC-2000 Utility on the *Driver and Manual CD*



Executing through the Command Line:

Execute the WD.EXE file under DOS (WD.EXE and CWSDPMI.EXE should be placed on same directory), then enter the values from 0~255. The system will reboot automatically according to the time-out you set.

//

You can write your own program by modifying the source code F81865_Test.cpp.. The index address is 2EH.

```
//////////////////////////////////////////////////////////////////////
/////
// F81865_Test.cpp : F81865_test.exe utility for F81865.lib
APIs demonstration.
//
// History:
//          7/15/2011          Brand new F81865_test
program.



#include <winsock2.h>
#include "Windows.h"
#include "stdio.h"
#include "F81865.h"



#define PARAMETER_HELP                         "\n"\
                    "The F81865 GPIO utility of Lanner\n"\
                    "--------------------------------------\n"\
                    "Usage:\n"\
                    "  F81865_test  DIO_IN           port_
number\n"\
                    "  F81865_test  DIO_OUT          port_
number  value\n"\
                    "  F81865_test  PIO              port_number
value\n"\
                    "  F81865_test  RunLED           port_
number  value\n"\
                    "  F81865_test  AlarmLED         port_
number  value\n"\
                    "  F81865_test  GPS_LED          port_
number  value\n"\
                    "  F81865_test  WirelessLED      port_
number  value\n"\

"  F81865_test  WatchDog           seconds\n"\

"  F81865_test  CaseOpen\n"\

"  F81865_test  CaseOpen_Clear\n"\
                    "  F81865_test  Sleep
milliseconds\n"\
                    "\n"\

"Argement:\n"\
                    "  DIO_IN           Read state from DIO
In.\n"\
                    "  DIO_OUT           Set DIO Out state.\n"\
                    "  PIO            Set PIO LED state.\n"\
                    "  RunLED           Set RUN LED state.\n"\

"  AlarmLED          Set Alarm LED state.\n"\

"  GPS_LED           Set GPS LED state.\n"\

"  WirelessLED        Set Wireless LED state.\n"\
            “ Watchdog                                  Set
Watchdog timer.\n”\

“ CaseOpen           Check case opened state.\n”\

“ CaseOpen_Clear       Clear case open state.\n”\
                  “ port_number         The port number.\n”\
                  “ value              1 for on and 0 for off.\n”\
                  “ seconds                                 The
watchdog count down seconds. 0 for disable.\n”\
                  “ milliseconds         Milliseconds to
delay\n”

#define RETMSG(a,b) {printf (b) ; return a;}

#define CHECK_ARGC(a) {if (argc != a) throw
PARAMETER_HELP ;}


// Translate Hex string to a long value
LONG Hex2Long (char *str)
{
            LONG nLong ;

            if (scanf (str, “%x”, &nLong) != 1)
                        throw “Error parsing parameter\n” ;

            return  nLong ;
}


// Make sure the argument is numeric
void CheckNumeric (char *szBuf)
{
            int nLen = strlen (szBuf) ;

            for (int i = 0 ; i < nLen ; i++)
      if (!strchr (“01234567890ABCDEFabcdef”, szBuf[i]) )
         throw “Wrong argument\n” ;
}


// Common GPIO output function definition
#define GPIO_OUT(a,b,c)                                  \
int a (int argc, char *argv[])                  \
{
                        \
            CHECK_ARGC (4) ;
\
            CheckNumeric (argv[2]) ;               \
            CheckNumeric (argv[3]) ;               \
```

```
                        \
            int nPort  = atoi (argv[2])  ;\
            int nValue = atoi (argv[3])  ;            \

                        \
            c (nPort, nValue) ;
\

                        \
            printf (b “ #%d = %d\n”, nPort, nValue) ;       \

                        \
            return 0 ;
            \
}


// Function generate by common function definition
GPIO_OUT (mDIO_OUT                  , “DIO_OUT”
, Write_DIO)
GPIO_OUT (mPIO                                  , “DIO_OUT”
            , PIO)
GPIO_OUT (mRunLED                   , “RunLED”
, RunLED)
GPIO_OUT (mAlarmLED                 , “AlarmLED”        ,
AlarmLED)
GPIO_OUT (mGPS_LED                  , “GPS_LED”
, GPS_LED)
GPIO_OUT (mWirelessLED              , “WirelessLED”     ,
WirelessLED)



// Check case open
int mCaseOpen  (int argc, char* argv[])
{
            CHECK_ARGC (2) ;


            BOOL bOpen = CaseOpen () ;
            printf (“Case is %s\n”, bOpen ? “Open” : “Close”) ;


            return bOpen ;
}
```

```
// Clear case open state
int mCaseOpen_Clear  (int argc, char* argv[])
{
        CHECK_ARGC (2) ;

        CaseOpen_Clear () ;

        BOOL bOpen = CaseOpen () ;

        printf (“CaseOpen state %s”, bOpen ? “not
cleared” : “cleared”) ;

        return bOpen ;
}


// Get DIO_IN state
int mDIO_IN (int argc, char* argv[])
{
        CHECK_ARGC (3) ;
        CheckNumeric (argv[2]) ;

        int  nPort = atoi (argv[2]) ;
        BOOL ret   = Read_DIO (nPort) ;

        printf (“DIO_IN #%d = %d\n”, nPort, ret) ;

        return ret ;
}


// Milli-second delay
int mSleep (int argc, char *argv[])
{
        CHECK_ARGC (3) ;

        CheckNumeric (argv[2]) ;

  Sleep (atoi (argv[2]) ) ;
```

```
        return 0 ;
}


// Watchdog
int mWatchDog (int argc, char *argv[])
{
        if (argc != 3 && argc != 2)
                RETMSG (-1, PARAMETER_HELP) ;

        if (argc == 3)
        {
                CheckNumeric (argv[2]) ;

                int nValue = atoi (argv[2])  ;

                WatchDog_Enable (nValue) ;
        }

        int nLeft = WatchDog_GetLeft () ;

        printf (“Watchdog timer left %d seconds\n”,
nLeft) ;

        return nLeft ;
}


// Argument - function mapping
typedef struct
{
        char *szCmd ;
        int (*function) (int argc, char *argv[]) ;

} CMD2FUN ;


CMD2FUN c2f[] =
{
        {"DIO_IN"               , mDIO_IN
},
        {"DIO_OUT"              , mDIO_OUT
},
        {"PIO"                  , mPIO
},
        {"RunLED"               , mRunLED
},
        {"AlarmLED"             , mAlarmLED
},
        {"GPS_LED"              , mGPS_LED
},
        {"WirelessLED"    , mWirelessLED    },
        {"CaseOpen"    , mCaseOpen    },
        {"CaseOpen_Clear",mCaseOpen_Clear},
        {"Watchdog"             , mWatchDog
},
        {"Sleep"          , mSleep          }
} ;


// Program start here
int main(int argc, char *argv[])
{
        try
        {
                // The total argument allowed
                int num = sizeof (c2f) / sizeof (c2f[0]) ;


                // Too few argument
                if (argc < 2)
                        RETMSG (-1, PARAMETER_
HELP) ;


                // Find the match argument and
execute the mapping function
                for (int i = 0 ; i < num ; i++)
                        if (stricmp (argv[1], c2f[i].
szCmd) == 0)
                                return c2f[i].function
(argc, argv) ;

                // No match argument
                RETMSG (-1, "Wrong Argument\n") ;
        }
        catch (char *str)
        {
                // Output the error message
                printf ("\n%s\n", str) ;
        }
        catch (...)
        {
                // Unknown exception
                printf ("\nUnknown Exception\n") ;
        }

    return -1 ;
}
```

Appendix E : Terms and Conditions

Warranty Policy

1. All products are under warranty against defects in materials and workmanship for a period of one year from the date of purchase.
2. The buyer will bear the return freight charges for goods returned for repair within the warranty period; whereas the manufacturer will bear the after service freight charges for goods returned to the user.
3. The buyer will pay for repair (for replaced components plus service time) and transportation charges (both ways) for items after the expiration of the warranty period.
4. If the RMA Service Request Form does not meet the stated requirement as listed on "RMA Service," RMA goods will be returned at customer's expense.
5. The following conditions are excluded from this warranty:

Improper or inadequate maintenance by the customer Unauthorized modification, misuse, or reversed engineering of the product Operation outside of the environmental specifications for the product.

RMA Service

Requesting a RMA#

6. To obtain a RMA number, simply fill out and fax the "RMA Request Form" to your supplier.
7. The customer is required to fill out the problem code as listed. If your problem is not among the codes listed, please write the symptom description in the remarks box.
8. Ship the defective unit(s) on freight prepaid terms. Use the original packing materials when possible.
9. Mark the RMA# clearly on the box.

Note: Customer is responsible for shipping damage(s) resulting from inadequate/loose packing of the defective unit(s). All RMA# are valid for 30 days only; RMA goods received after the effective RMA# period will be rejected.

RMA Service Request Form

When requesting RMA service, please fill out the following form. Without this form enclosed, your RMA cannot be processed.

RMA No:

Reasons to Return: □ Repair(Please include failure details)
□ Testing Purpose

Company:

Contact Person:

Phone No.

Purchased Date:

Fax No.:

Applied Date:

Return Shipping Address: ____

Shipping by: □ Air Freight □ Sea □ Express____________
□ Others:________________

Item	Model Name	Serial Number	Configuration

Item	Problem Code	Failure Status

*Problem Code:

01:D.O.A.
02: Second Time R.M.A.
03: CMOS Data Lost
04: FDC Fail
05: HDC Fail
06: Bad Slot
07: BIOS Problem
08: Keyboard Controller Fail
09: Cache RMA Problem
10: Memory Socket Bad
11: Hang Up Software
12: Out Look Damage
13: SCSI
14: LPT Port
15: PS2
16: LAN
17: COM Port
18: Watchdog Timer
19: DIO
20: Buzzer
21: Shut Down
22: Panel Fail
23: CRT Fail
24: Others (Pls specify)

Request Party

Confirmed By Supplier

____

Authorized Signature / Date

____

Authorized Signature / Date