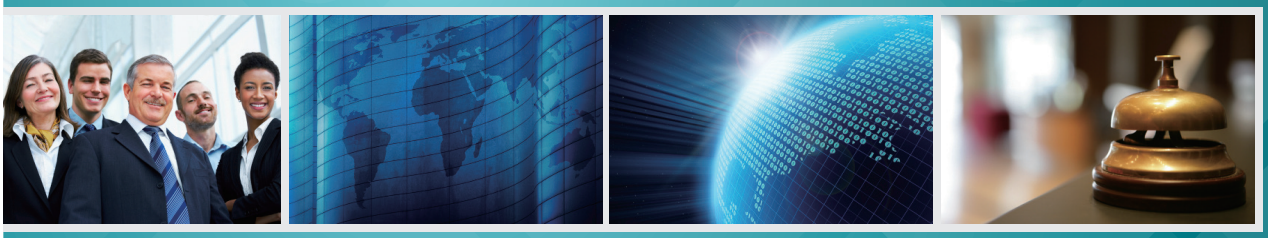
Lanner

creating value in applied computing

In-Vehicle Computing

Hardware Platforms for mobile applications



LVC-2000
Preliminary

User's Manual

Publication date:2014-10-13

About

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

Acknowledgement

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

About

Mechanical compliance

Low Pressure (Altitude):

- MIL-STD-810G, Method 500.5, Procedure I (Storage)
- MIL-STD-810G, Method500.5, Procedure II (Operational)

High Temperature:

- MIL-STD-810G, Method 500.5, Procedure I (Storage)
- MIL-STD-810G, Method500.5, Procedure II (Operational)

Low Temperature:

- MIL-STD-810G, Method 500.5, Procedure I (Storage)
- MIL-STD-810G, Method500.5, Procedure II (Operational)

Temperature Shock:

MIL-STD-810G, Method 503.5, Procedure I-C

General Vibration (Operating):

MIL-STD-810G, Method 514.6, Procedure I, Category4, Figure 514.6C-1 (Common Carrier, US Highway Truck Vibration Exposure)

General Vibration (Non-operating):

MIL-STD-810G, Method 514.6, Procedure I, Category 24, Figure 514.6E-1 (General Minimum Integrity)

Shock (Operating):

MIL-STD-810G, Method 516.6, Procedure I (Functional Test for Ground Equipment)

Shock (Non-operating):

MIL-STD-810G, Method 516.6, Procedure V (Crash Hazard Shock Test for Ground Equipment)

Transit Drop:

MIL-STD-810G, Method 516.6, Procedure IV (Transit Drop)

Revision History

Table of Contents

Chapter 1: Introduction

Thank you for choosing the LVC-2000.

The LVC-2000 is a compact in-vehicle computing system designed with support for wall mount 1 and it has a vibration kit built in for eliminating shock and vibration, ideal for deployment and installation on moving transportation vehicles.

The system is equipped with the newest 22nm process Intel® Atom™ processor E3845 family (formerly codenamed Bay Trail). This system-on-chip (SoC) incorporates up to four cores and supports system memory DDR3L-1066/1333 SO-DIMM maximum up to 8 GB. This small form factor is capable of operating under wide temperatures from -20°C to +60°C in fanless operation, which allows the system to operate in rugged environments. Moreover, it comes with multiple display outputs: HDMI and VGA. With integrated Intel® Gen HD graphics, visual capabilities are enhanced, including faster media processing, full HD over HDMI, spectacular HD playback, etc.

The system encompasses a wide variety of communication ports to facilitate every possible in-vehicle applications:

- One Ethernet port provided by Intel i210IT Ethernet controller.
- Two additional digital input pins from the Multiple (MIO) I/O port can be used for system wake-up to power on the system automatically; another two digital output pins from the same Multiple I/O (MIO) port can be used for control relay (current @2mA)
- Multiple I/O ports for Digital I/O and serial port connections
- Rich I/O ports: two RS-232, 3 USB ports (one USB 3.0 type A, 2 USB 2.0 with pin headers)
- Three Mini-PCIe connectors (one supports mSATA; one full-size with SIM card reader for 3G wireless Internet connection; one half-size for Wi-Fi connection)
- Dual video display: HDMI+VGA output with Intel integrated HD graphic engine
- Power ignition control mechanism with programmable on/off/delay switch
- Wide range of DC power input from 9V to 36V, suitable for vehicular 12V or 24V battery with Ignition control.

  –Power input current protection by 15KP30A TVS

  --12V DC output current with a maximum of 1A
- Battery voltage protection: Over Voltage Protection and Under Voltage Protection

System Specifications

Dimensions (WxHxD)		210 x 52 x 144 mm
Processor		ntel Atom processor E3845 1.9GHz (Option for E3815 / E3825 / E3826 / E2827)
System Memory	Technology	DDR3L SO-DIMM x1 (Factory default: 2GB module pre-installed)
	Max. Capacity	Up to 8GB (user option)
Storage	mSATA/SATA	Removable 2.5" SSD/HDD drive bay x1, Mini-PCIe connector for mSATA
Ethernet Controller		Intel i210IT
Graphic Controller		Intel integrated HD graphic engine
Audio Controller		Realtek ALC886-GR
IO	LAN	GbE RJ45 x1
	Display	VGA, maximum resolution up to 2048x1536@60Hz HDMI, maximum resolution up to 1920x1200@75Hz
	Audio	Internal pin header for Mic-in and Line-out
	Serial I/O	1 x RS-232 1 x RS-232/422/485
	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 DO control Relay support 9~36V@max 2A each 2x DI to Ignition MCU as remote control 1x 12VDC Out 1x Rx/Tx
	USB 2.0	USB 3.0Type A x 1 USB 2.0 Pin headers x2
	Power Input	3-pin terminal block (+, -, ignition)
	Expansion	Mini-PCIe x2 (one is full-size with SIM card reader; the other one is half-size)
	CAN bus	supports J1939 & J1708
Power Input		+9~36VDC input range, with ignition delay on/off control
OS Support		Windows7/ 7 Embedded /8 embedded OS Image: WES7 (64bit & 32bit) / W7 FES (64bit & 32bit) / Windows 8(32bit); Linux kernel 2.6.X or later
Certifications		CE, FCC Class A, E13, RoHS

Compliance		**Low Temperature:** MIL-STD-810G, Method 502.5, Procedure I (Storage) MIL-STD-810G, Method 502.5, Procedure II (Operational) **Temperature Shock:** MIL-STD-810G, Method 503.5, Procedure I-C **General Vibration (Operating):** MIL-STD-810G, Method 514.6, Procedure I, Category4, Figure 514.6C-1 (Common Carrier, US Highway Truck Vibration Exposure) **General Vibration (Non-operating):** MIL-STD-810G, Method 514.6, Procedure I, Category 24, Figure 514.6E-1 (General Minimum Integrity) **Shock (Operating):** MIL-STD-810G, Method 516.6, Procedure I (Functional Test for Ground Equipment) **Shock (Non-operating):** MIL-STD-810G, Method 516.6, Procedure V (Crash Hazard Shock Test for Ground Equipment) **Transit Drop:** MIL-STD-810G, Method 516.6, Procedure IV (Transit Drop)
Operating Temperature Range	Extended	With Selected Industrial Components -20~60°C/-4~140°F

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)

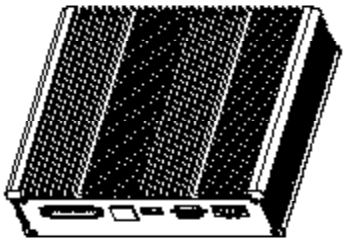
- HDD Screws x 4 (P/N: 070W102400602)
- Mini-PCIe Screws x 4 (P/N: 070W101000401)

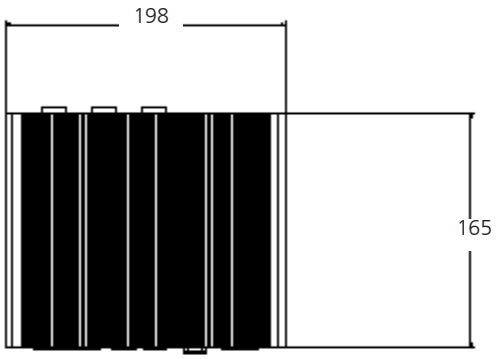
Chapter 2: System Components

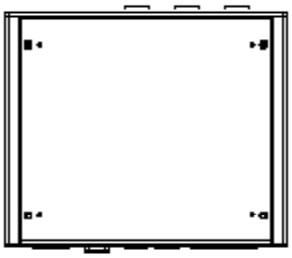
System Drawing

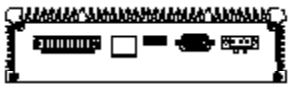
Mechanical dimensions of the LVC-2000 with the wall mount kit (suspension kit).

Unit: mm



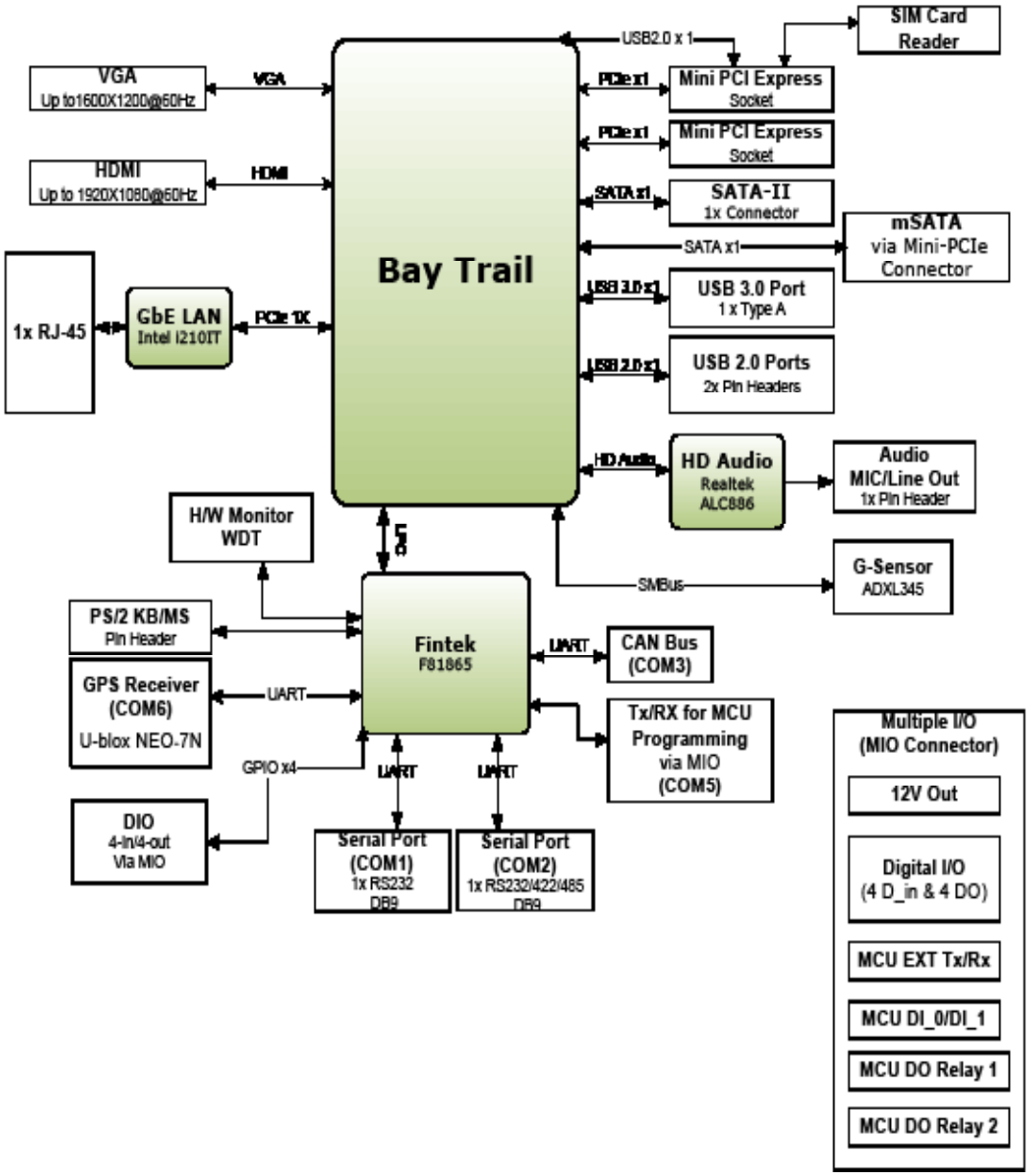




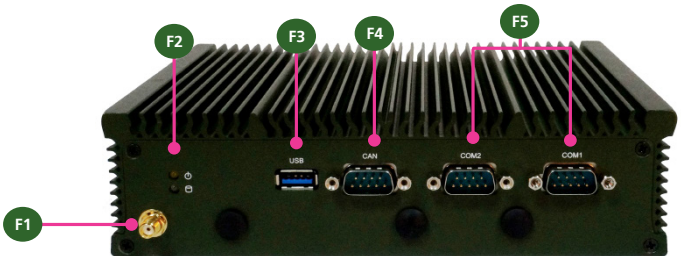


Block Diagram: The MainBoard

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

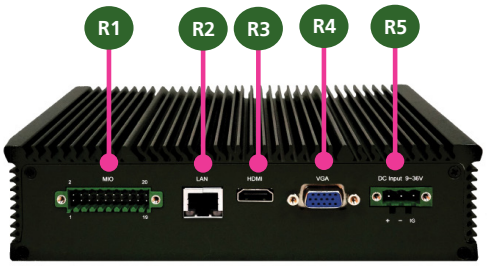


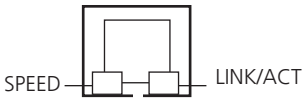
Front Components



Component	Description	Pin Definition Reference
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. There are additional 2 USB 2.0 ports with pin headers	USB2 on page
F4 CAN bus	CAN bus connector for controller area network communication. It supports J1939 &J1708 standards.	CAN1 on page
F5 COM1/COM2	RS232 ports for serial communication	COM1/COM2 on page

Rear Components



Component	Description	Pin Definition Reference
R1 Multiple-I/O Connector	A 20-pin male connector for the following functions: • 4 Digital-In & 4 Digital-output • 12VDC power output • Two Output relay control with contact current which support 9~36V@ 2A each • MCU input detection to wake up the system automatically • One serial communication port	MIO2 on page 20
F3 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.	
R2 HDMI Port (‡)	A HDMI port which is provided by Intel HD graphics (resolution: 1920x1080@60Hz). There is also an internal Audio pin header for HD Audio MIC-in/Line-out	HDMI1 on page 19
R4 VGA Port (‡)	It connects an external VGA monitor or projector (resolution: 1600x1200@60Hz)	VGA1 on page 19
R6 Power-In (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.	PRJK1 on page 21

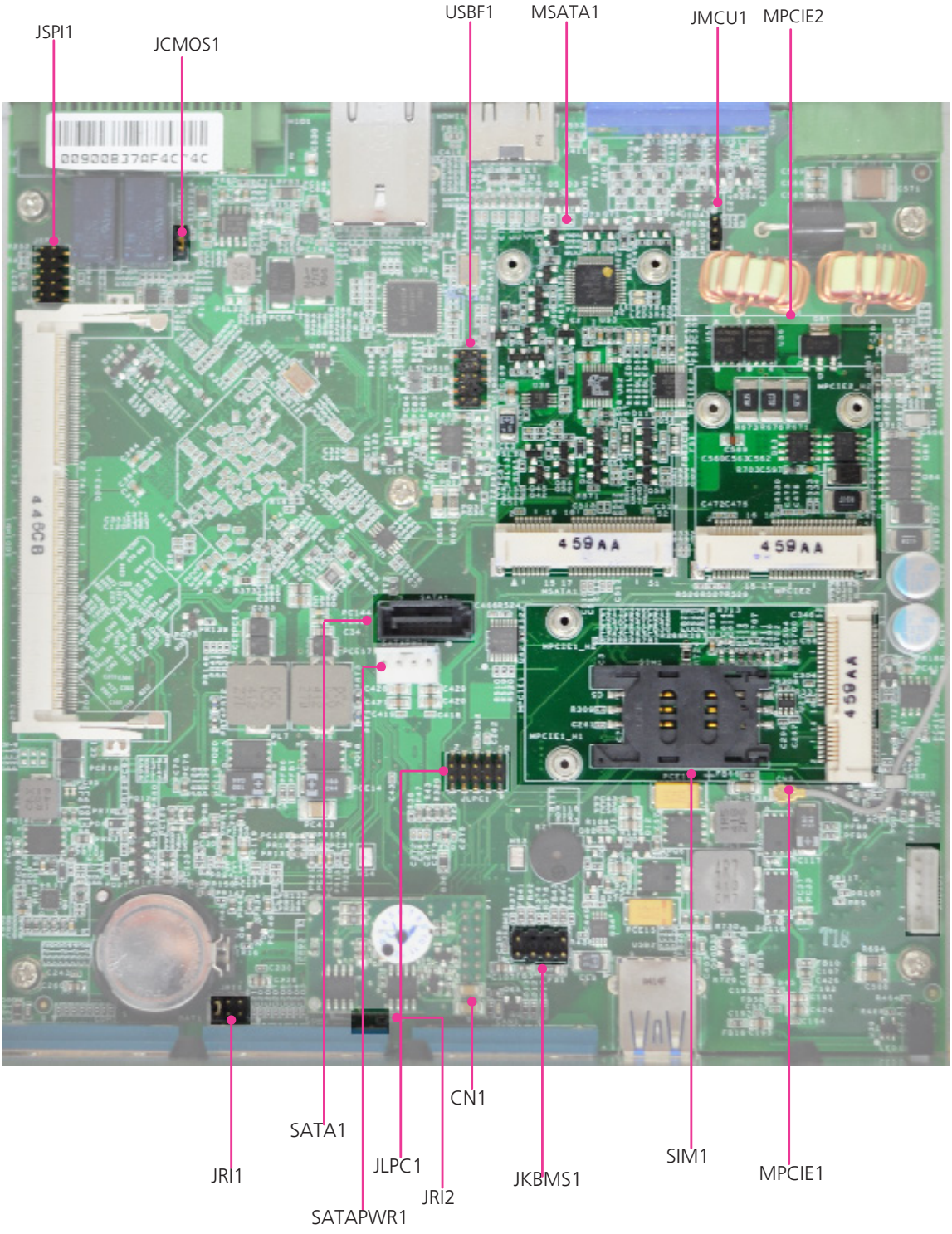
Chapter 3: Board Layout

External Connectors

The following picture highlights the location of internal connectors and jumpers. Refer to the table 3.1 Connector List for more details.

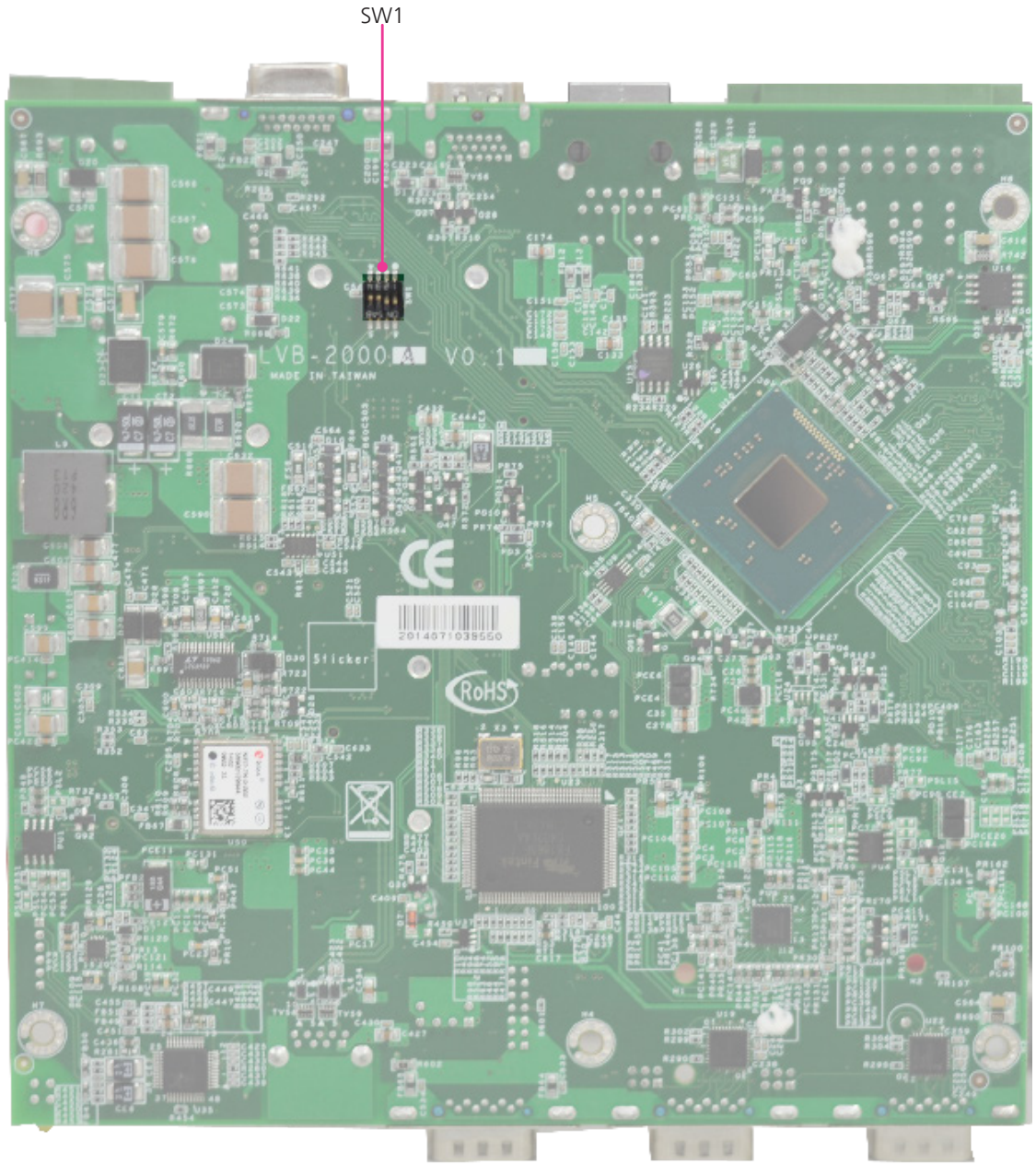
Internal Connectors and Jumpers

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

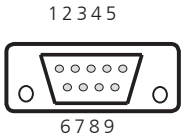
Labels	**Function**	**Pin Definition Reference Page**
CAN1	CAN bus Connector	
COM1/COM2	RS-232 Commmunication Ports	P
HDMI1	High Definition Multimedia Interface	P
MIO1	Multiple I/O Connectors	
PRJK1	3-Pin DC-in Power Connector with Ignition Control	
USB2	USB 3.0 Connector	P
VGA1	VGA Connector	

Table 3.2 Connector List for Internal Connectors

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

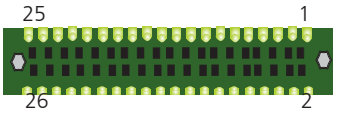
Jumper Settings

COM1 RS-232 Serial Port (COM1/COM2): An RS-232 port through the D-SUB9 connector. The RI (pin 8) can be altered with jumper JRI1 and JRI2.



Pin No.	Pin Name
1	DCD
2	DSR
3	RXD
4	RTS
5	TXD
6	CTS
7	DTR
8	RI1
9	GND

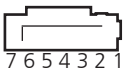
Multiple I/O Connectors (MIO1): Multiple I/O pins for functions in serial communication, Digital In/Out, Ignition detection input for automatic wake-up function



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).			

Serial-ATA Connector (SATA1): It is for connecting a 2.5" hard disk to be served as your system's storage. It can supports SATA 2.0.



SATA1

Pin No.	Function
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 No.	Pin Name
1	VCC12
2	GND
3	GND
	VCC5

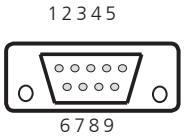
AUDIOIN1: Line-in and Mic-in Connector

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

MCU Programming Connector (JMCU1)

Pin No.	Pin Name
1-2	Program MCU
2-3	Normal

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

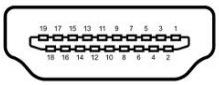
MPCIE1: Mini-PCIe Connector with one SIM Card Reader(SIM1). It supports both Wi-Fi and 3G module.

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)

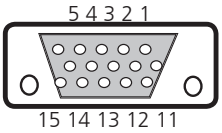
Pin	Signal	Pin	Signal
1	PCIE_WAKE_N	2	VCC3P3_PS
3	N/A	4	GND
5	N/A	6	V1P5_MPCIE
7	E_CLKREQ-	8	UIM2_PWR
9	GND	10	RSV
11	PCIE_CKN4	12	RSV
13	PCIE_CKP4	14	RSV
15	GND	16	RSV
17	RSV	18	GND
19	RSV	20	N/A
21	GND	22	BUF_PLT_RST#
23	PCH_PCIE_RXN4	24	PCIE_PCIE_VCC3AUX
25	PCH_PCIE_RXP4	26	GND
27	GND	28	V1P5_MPCIE
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

HDMI Connector (HDMI1): An HDMI Connector



Pin No.	Description	Pin No.	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	PHDMI
19	HDMI_HPD		

VGA (VGAA1)



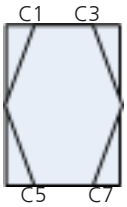
Pin	Signal	Pin	Signal	Pin	Signal
1	RED	6	GND	11	N/A
2	GREEN	7	GND	12	DDC DAT
3	BLUE	8	GND	13	HSYNC
4	N/A	9	VCC5	14	VSYNC
5	CRT_DET	10	GND	15	DDC CLK

Note: The driver for the VGA and Audio ports should be installed with the following order: Chipset INF->Graphic->Audio

mSATA (MSATA1) for mini-SATA Connector: It complies with SATA 2.0

PIN	Pin Name	PIN	Pin Name
1	N/A	30	SMB_CLK
3	V3P3S	31	mSATATXN
5	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 Socket (SIM1): SIM1 pairs with MPCIE1 and SIM2 pairs with MPCIE2.



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

Power-in with Ignition Control (PRJK1): A power connector with power -ignition Control

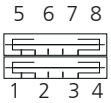
Pin No.	Pin Name
1	Ignition
2	GND
3	DC_VIN

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.

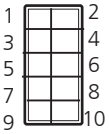
Pin No.	Signal
1-2	Default
3-4	VCC5
5-6	VCC12

USB 2.0 Pin Header for USB 0, USB1 (USBF1)

USB 3.0 Port Connector for USB2 (USB2)



Pin No.	Pin Name
1	VCCUSB2
2	USB0N2
3	USB0P2
4	GND
5	USB3_SSRXN
6	USB3_SSRXP
7	GND
8	USB_SSTXP
9	USB_SSTXN

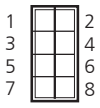


Pin No.	Pin Name
1	VCC5
2	GND
3	N/A
4	USBDDP1
5	USBDN0
6	USBDN1
7	USBDP0
8	N/A
9	GND
10	VCC5

Clear CMOS (JCMOS1):

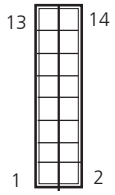
Pin No.	Pin Name
1-2	Normal (Default)
2-3	Clear RTC

PS/2 Keyboard and Mouse Connector (JKBMS1)



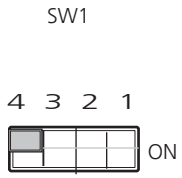
Pin No.	Pin Name	Pin No.	Pin Name
1	VCC	2	MCLK
3	MDATA	4	NC
5	KDATA	6	NC
7	GND	8	KCLK

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-

Select MCU Detect Function for power ignition behavior (SW1):



Selector No.	SW1	Ignition Function
1	Power Good Detection	ON: Enable OFF: Disable
2	Low Voltage Detection	
3	Watchdog	
4	Programming MCU	Reserved
The default value is ON for selector 1, ON for selector 2, OFF for selector 3, and OFF for selector 4		

The functions of the above jumpers are further explained here.

1. **Power Good Detection**: A power-good signal from the main board will be sent to the ignition controller so that the ignition controller can decide or alter the power state upon the following instances. (Refer to the ***flow chart*** in **Chapter 4**):
   - Power-on instance
   - Power-good signal turned-low instance
2. **Low Voltage Detection**: Turn on this switch to enable the automatic detection of low voltage state of the battery. It will automatically turn off the system when low voltage state has been detected (**Note**: **the low-voltage condition needs to remain 30 seconds continually**). The voltage level can be set in the *Ignition System Manager* (ISM) which is provided by Lanner as a sample code for functions on the power ignition module. **The default setting of this function: Shutdown Voltage in the ISM is disabled.** (Refer to the ***flow chart*** in **Chapter 4** and the ***Using the Ignition System Manager (ISM)*** in **Appendix A**.)
3. **Watchdog:** Enable this switch to enable shutdown after watchdog timer count-down to zero. This is a programmable function. If there is no program to control and monitor the watchdog timer, set this jumper to disabled to avoid abnormal shutdown. The default time-out value is 300 sec(you will need an AT command to reset watchdog timer; contact Lanner rep for this program).

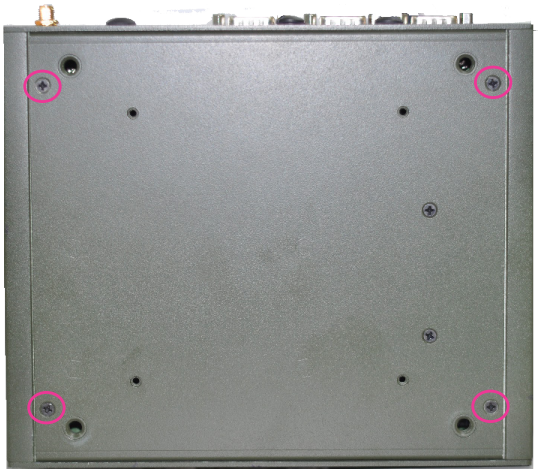
Chapter 4: Hardware Setup

Preparing the Hardware Installation

To access some components and perform certain service procedures, you must perform the following procedures first.

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 remain active until power is removed.

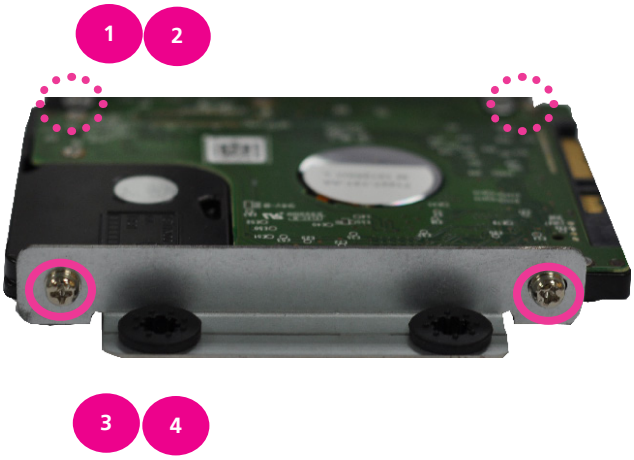
1. Unpower the LVC-2000 and remove the power cord.
2. Remove 4 threaded screws from the bottom to take off the bottom cover.
3. Open the cover.



HDD Installation

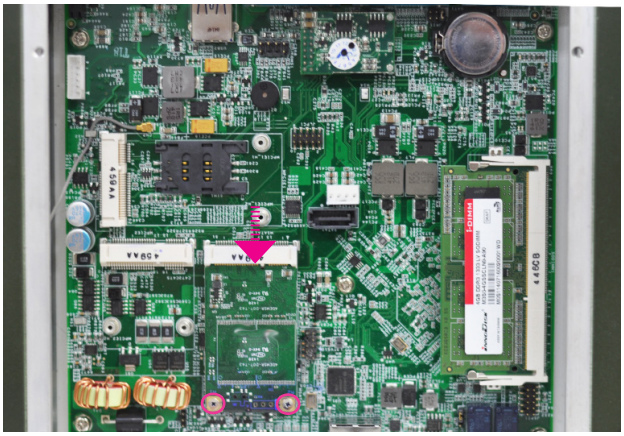
The system can accommodate one Serial-ATA disk. Follow these steps to install a hard disk into the system:

1. Take out the hard disk tray and fix the hard disk on the tray with 4 mounting screws as illustrated in the following picture.
2. Plug the Serial-ATA cable to the hard disk.
3. Place the hard disk back to the system's chassis and fix 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.

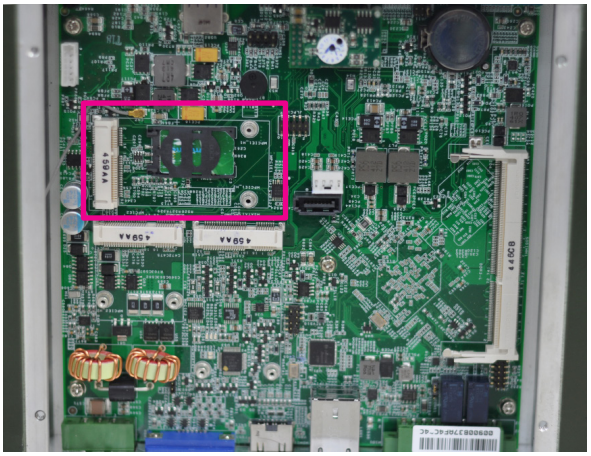


Wireless Module Installation

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.

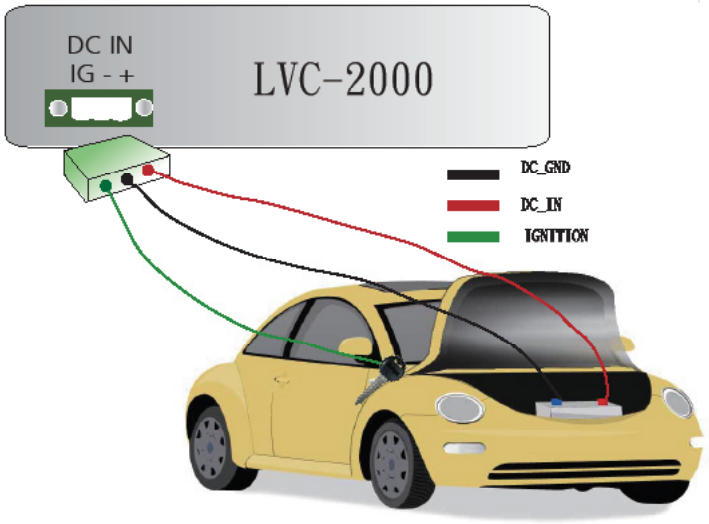
3G SIM Card Installation

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.



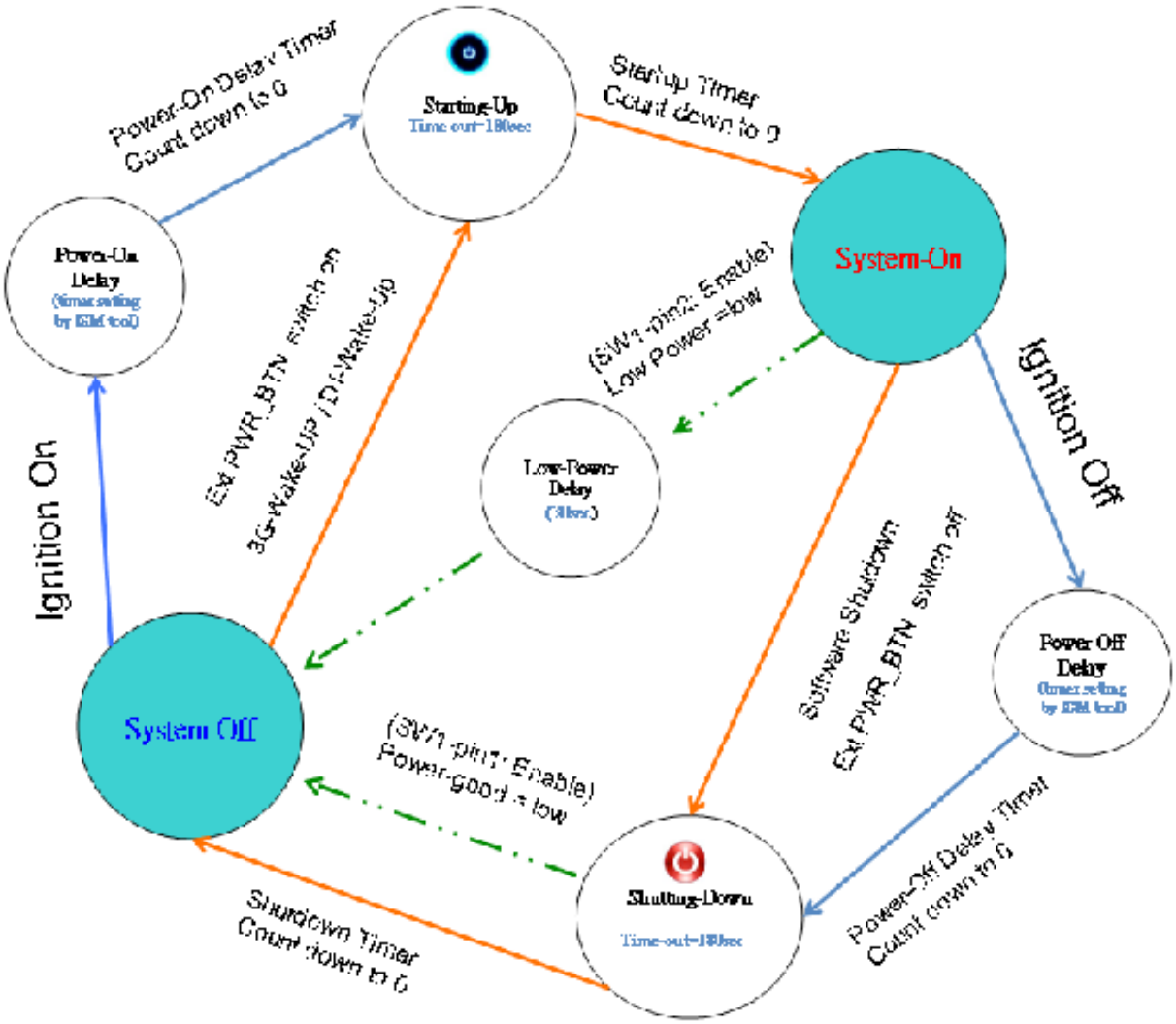
Connecting Power

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 when inserting power to the system.



Chapter 5: 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 MIO1 ***Pin 3*** and ***4***. 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 starts counting down 180sec, using ignition or External PWR_BTN to start the system again during shutdown process will not work until the countdown finishes.

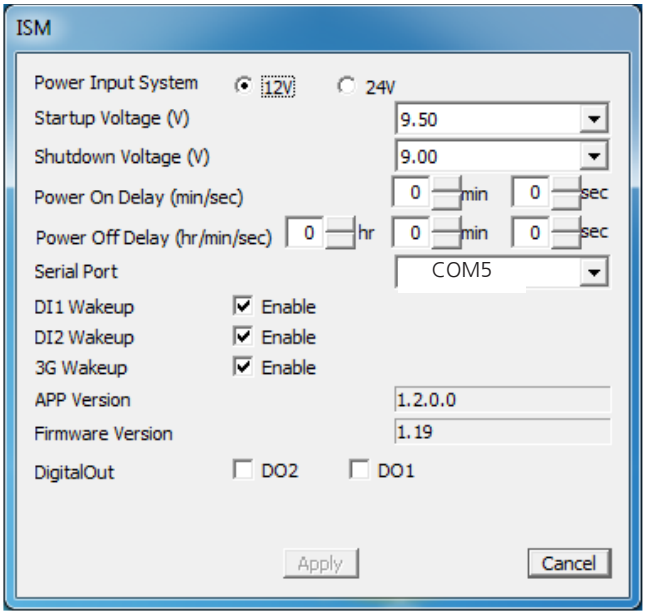
Appendix A: Using the Ignition System Manager (ISM)

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.

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 3 and 4 of MIO1 (Refer to MIO1 in **Chapter 3 Board Layout**)
3. Both DO1 and DO2 functions are connected (controlled) via pin 15, 16, 19 and 20 of MIO1. (Refer to MIO1 in **Chapter 3 Board Layout**)
4. Refer to the flow charts in Chapter 4 for more information.

.

Appendix B: Digital Input/Output

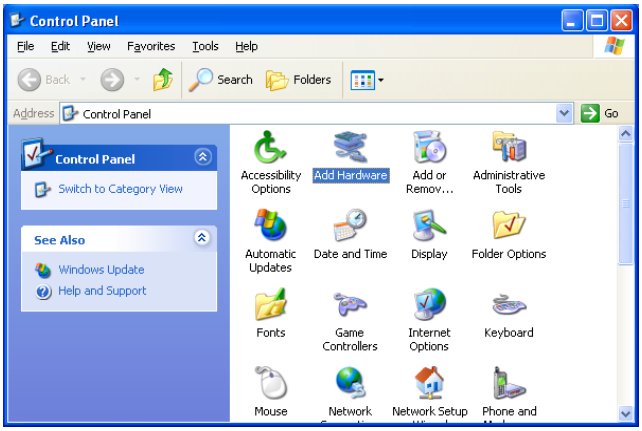
The Digitanl I/O on the rear panel is designed to provide the input and output operations for the system. For sample DIO code, see SuperIO folder under LVC-2000 Utility on the *Driver and Manual CD*. Make sure that you have installed the Lanner GPIO driver as instructed below.

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.

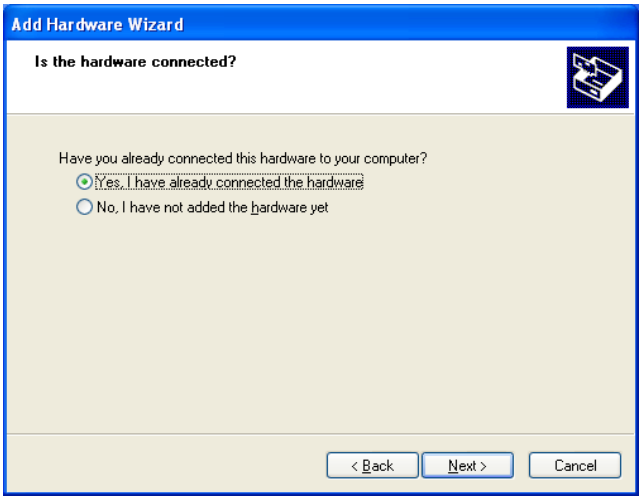
To install the L_IO driver:

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

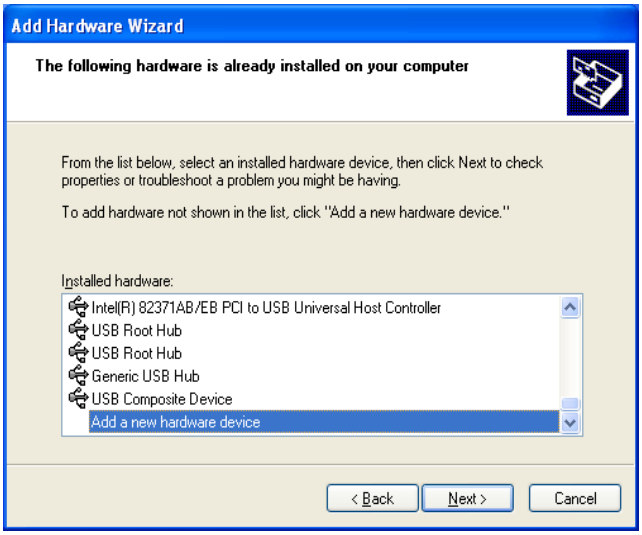


5. Select Next to proceed

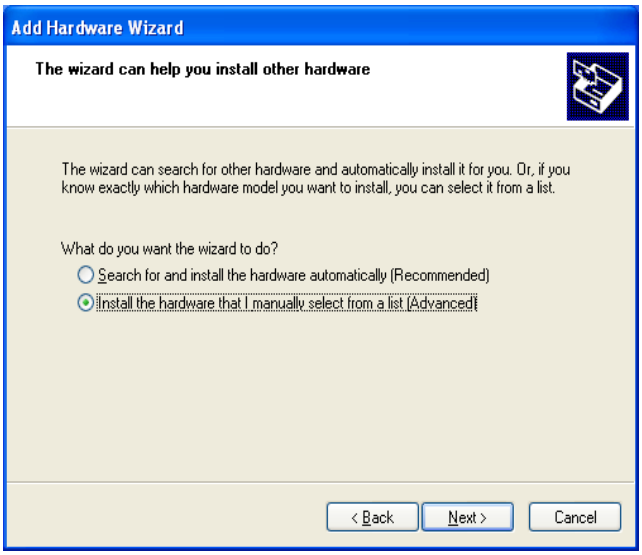
6. Answer "Yes" to the question and select Next to proceed.



7. Select Add a new hardware device.

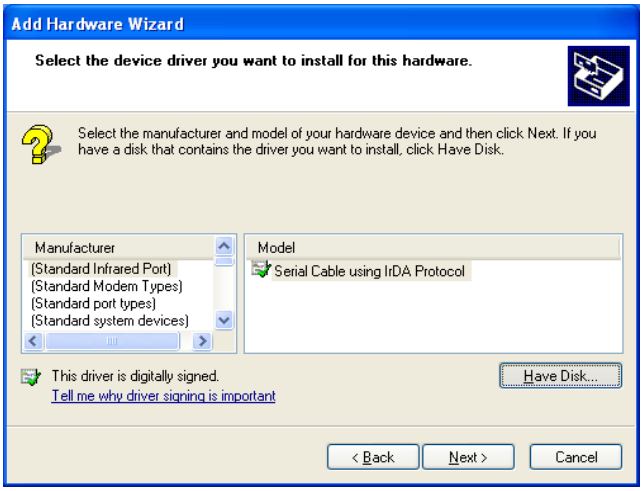


8. Choose to select the hardware Manually

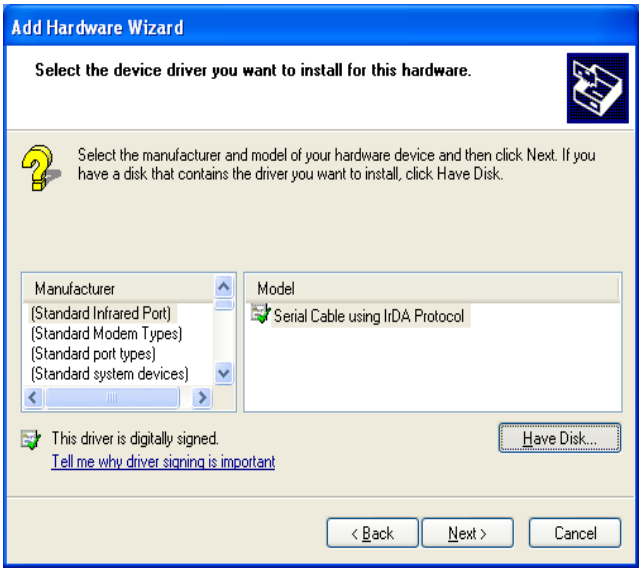


9. Choose Show all device and click Next.

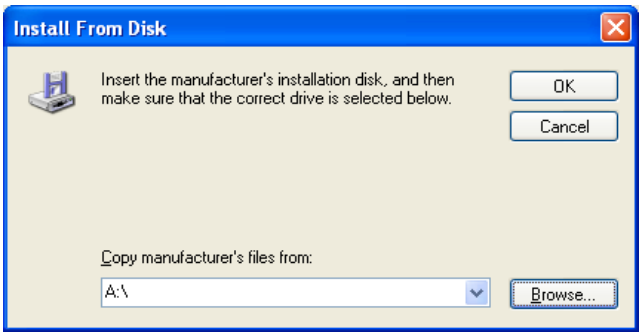
10. Click HaveDisk to locate the L_IO.inf file



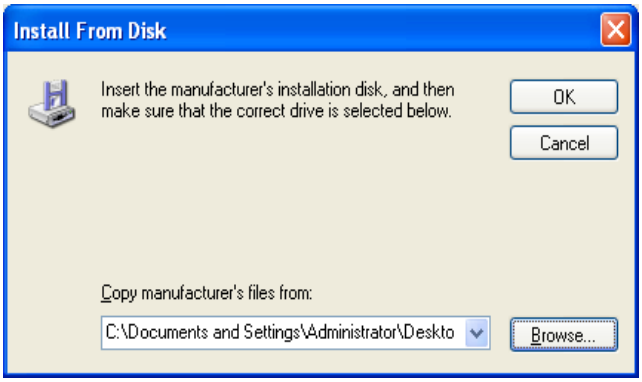
11. Click HaveDisk to locate the L_IO.inf file



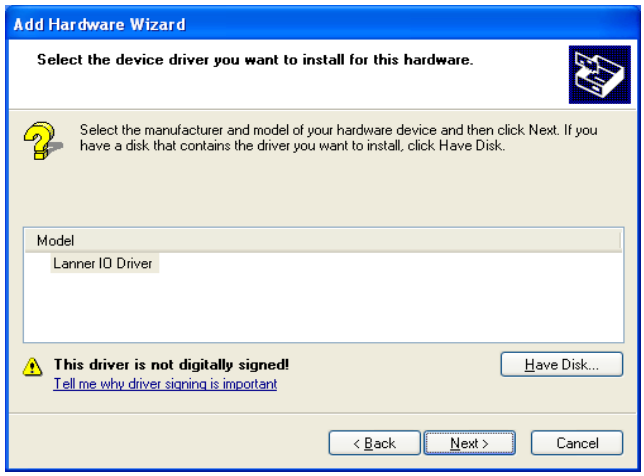
12. Select the L_IO.inf



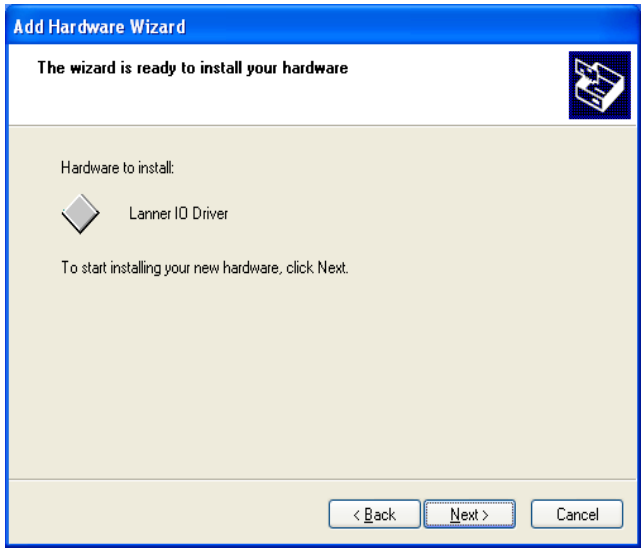
13. Select OK to confirm with the installation



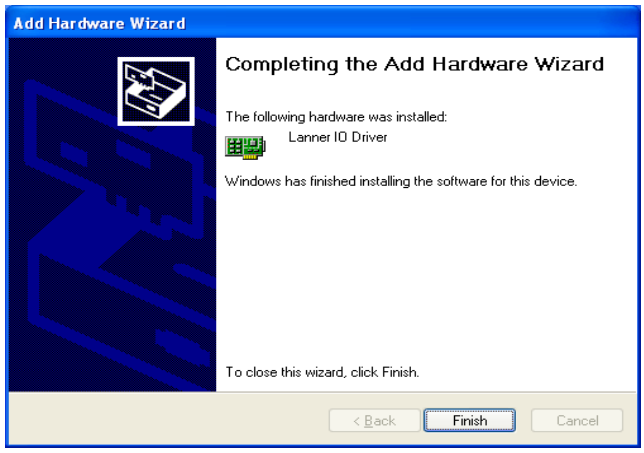
14. Select the Lanner IO driver and click Next.



15. Click Next



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



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.

Appendix B

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>

Appendix B

Digital Input/Output Control

```
#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 Digital Accelerometer Data from the LVC-2000

The system employs Analog Devices's ADXL345 Digital Accelerometer which is a small, thin, ultralow power, 3-axis accelerometer with high resolution (13-bit) measurement at up to ±16 g. It interfaces with the LVC-2000 through a SPI interface.

To access the Gsensor data, locate the adxl345_v001 folder and execute the executable file adxl345 and it will show G value of 3 axes.

Driver Installation

To access the G-Sensor data, use the following instructions:

1. Make sure you already installed the Lanner GPIO driver on your LVC-2000 as instructed in Appendix B.
2. To access the Gsensor data, locate the adxl345_v001 folder and execute the executable file adxl345 and it will show G value of 3 axes.

A sample program in C:++

```
// main.cpp
// The adxl345.exe utility shows the 3 axis G value.
//
// History:
//          07/15/2011: Initial version

#include <winsock2.h>
#include <windows.h>
#include <stdio.h>
#include "ich7.h"
#include "adxl345.h"


void adxl345_init()
{
    ich7_SM_WriteByte (0x1D, POWER_CTL, ACT_INACT_SERIAL | MEASURE) ;                    // Power CTL: Measure mode, Activity and Inactivity Serial


  ich7_SM_WriteByte (0x1d, BW_RATE, RATE_100);
                                                                                        /     /
Output Data Rate: 100Hz


      ich7_SM_WriteByte (0x1d, DATA_FORMAT, FULL_RESOLUTION | DATA_JUST_LEFT | RANGE_16G);



                                                                                        /     /
Data Format: 16g range, right justified, 256->1g
}



int main(int argc, char* argv[])
{
          adxl345_init () ;


          while (1)
          {
                    short x = (short) ich7_SM_ReadByte (0x1d, DATAX1) << 8 | ich7_SM_ReadByte (0x1d, DATAX0)<<0 ;
                    short y = (short) ich7_SM_ReadByte (0x1d, DATAY1) << 8 | ich7_SM_ReadByte (0x1d, DATAY0)<<0 ;
                    short z = (short) ich7_SM_ReadByte (0x1d, DATAZ1) << 8 | ich7_SM_ReadByte (0x1d, DATAZ0)<<0 ;


                    printf ("\rX=%.2f Y=%.2f Z=%.2f", ((float) x)/2048,((float)y)/2048,((float)z)/2048) ;
          }
}
```

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

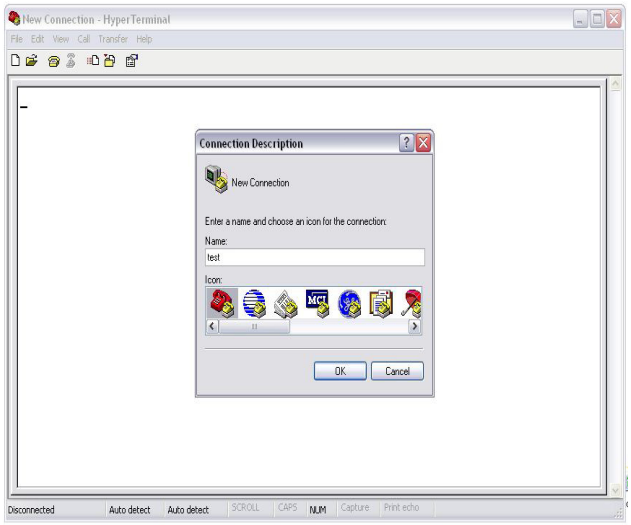
The LVC-2000 employs an onbard u-blox NEO-7N GPS module for vehicle tracking and navigation system. You could read the GPS data through the RS-232 serial port.

It has the following listed key features and performance ratings:

Receiver type	56-channel u-blox 7 engine GPS L1 C/A, GLONASS L1 FDMA QZSS L1 C/A Galileo E1B/C SBAS: WAAS, EGNOS, MSAS
Navigation update rate	up to 10 Hz
Accuracy GPS / GLONASS	Position 2.5 m CEP / 4.0 m SBAS 2.0 m CEP / n.a.
Acquisition GPS / GLONASS	Cold starts: 29 s / 30 s Aided starts: 5 s / n.a. Reaquisition: 1 s / 3 s
Sensitivity GPS / GLONASS	Tracking: –162 dBm / –158 dBm Cold starts: –148 dBm / –140 dBm Warm starts: –148 dBm / –145 dBm
Assistance	AssistNow Online AssistNow Offline AssistNow Autonomous OMA SUPL & 3GPP compli-ant
Oscillator	TCXO (NEO-7N), crystal (NEO-7M)
RTC crystal	built-in
Anti jamming	active CW detection and removal
Memory	ROM (NEO-7M) or Flash (NEO-7N)
Supported antennas	active and passive

To access the GPS data, follow the following steps:

Select Programs from the Start menu on your windows and open the Hyper Terminal program.



Choose COM4 from the Connection using drop-down menu:

Appendix D

Accessing the GPS Data

Specify the following communication parameters:

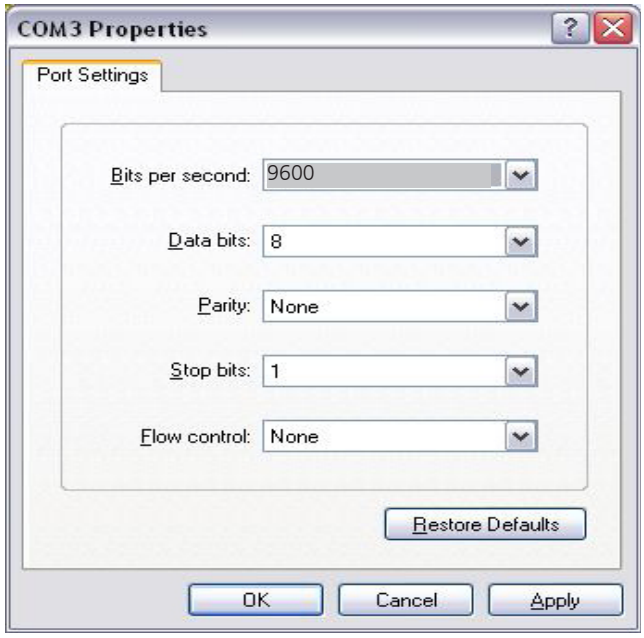
Bits per Second: 9600

Data Bits: 8

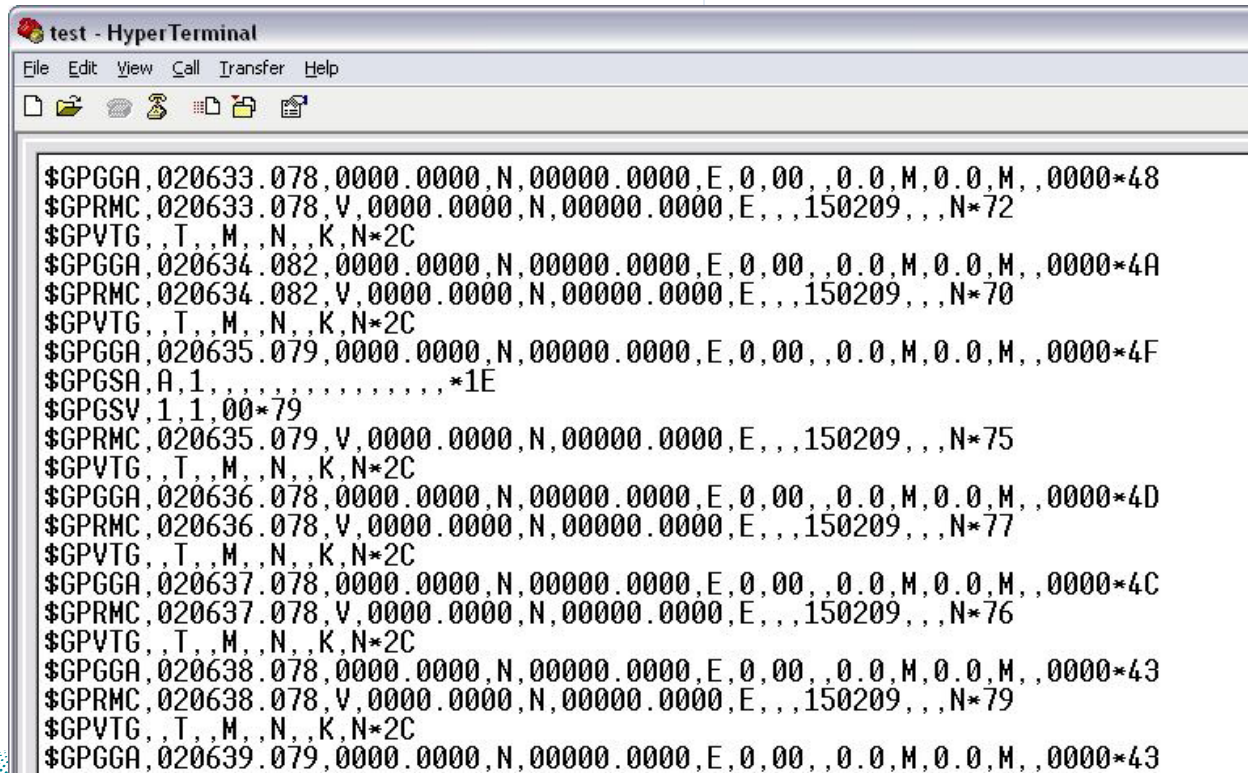
Parity: None

Stop Bit: 1

Flow Control: None



The hyper terminal should display GPS data:



Appendix E: 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)
G     P     I     O     _     O     U     T
(mPIO                 , "DIO_OUT"             , PIO)
G     P     I     O     _     O     U     T
(mRunLED              , "RunLED"              , RunLED)
G     P     I     O     _     O     U     T
(mAlarmLED            , "AlarmLED"      ,   AlarmLED)
GPIO_OUT                                    (mGPS_
LED             , "GPS_LED"             ,   GPS_LED)
G     P     I     O     _     O     U     T
(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
(argv, 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 B: 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.

Appendix B

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

Request Party *Confirmed By Supplier*

_____ _____

Authorized Signature / Date Authorized Signature / Date