DB-8151

Modulized Function Board - HSL

User's Manual

Manual Rev. 2.00

Revision Date: May 20, 2008

Part No: 50-14009-2000

Recycled Paper

Advance Technologies; Automate the World.

The information in this document is subject to change without prior notice in order to improve reliability, design, and function and does not represent a commitment on the part of the manufacturer.

In no event will the manufacturer be liable for direct, indirect, special, incidental, or consequential damages arising out of the use or inability to use the product or documentation, even if advised of the possibility of such damages.

Trademarks

NuDAQ, NuIPC, DAQBench are registered trademarks of ADLINK TECHNOLOGY INC.

Product names mentioned herein are used for identification purposes only and may be trademarks and/or registered trademarks of their respective companies.

Getting Service from ADLINK

Customer Satisfaction is top priority for ADLINK Technology Inc. Please contact us should you require any service or assistance.

ADLINK TECHNOLOGY INC.

Web Site:	http://www.adlinktech.com
Sales & Service:	Service@adlinktech.com
TEL:	+886-2-82265877
FAX:	+886-2-82265717
Address:	9F, No. 166, Jian Yi Road, Chungho City, Taipei, 235 Taiwan

Please email or FAX this completed service form for prompt and satisfactory service.

Company Information	
Company/Organization	
Contact Person	
E-mail Address	
Address	
Country	
TEL	FAX:
Web Site	
Product Information	
Product Model	
Environment	OS: M/B: CPU: Chipset: BIOS:

Please give a detailed description of the problem(s):

Table of Contents

List of Tables

List of Figures

1 Introduction

In order to increase the available functions on the PCI-8154/58 4-axis/8-axis pulse train output control channel carrier board, ADLINK offers four daughter boards providing a variety of functions. These DB-815x daughter boards can only be used with the PCI-8154/58 carriers only. Additional functionality provided by the DB-815x series includes high-speed triggering and distributed I/O control. Daughter boards can be added based on application requirements. The four daughter boards in the DB-815x series are:

Model Name	Primary Function	Description
DB-8150	High Speed Trigger Out	▶ High speed trigger pulse out up to 1 MHz ▶ Simultaneous 8 differential trigger output ▶ 2 channels encoder input (from external I/F or carrier. FIFO/Linear function compared trigger output
DB-8151	HSL Master Controller	▶ High Speed Link for distributed I/O
DB-8152	ECAM Controller	▶ 2 channel encoder input (Master & slave) ▶ Master/Slave controller ▶ Easy to implement electronic cam behavior
DB-8153	Motionnet Master Controller	▶ Serial single axis motion control bus ▶ Does not support I/O function

Table 1-1: DB-815x Series

The DB-8151 offers an HSL (High Speed Link) system for distributed I/O on the PCI-8154/58. HSL is an innovative distributed I/O

technology which allows thousands of I/O points to be scanned at the millisecond-level in real time by using mater-slave architecture. Commercial Ethernet cables with RJ45 connector are used for simplified setup of the HSL slaves modules as close as possible to sensor devices, again, resulting in a dramatic reduction of wiring. System integrators can benefit from HSL network because it integrates discrete I/O, and analog I/O. This local network features rapid response, real-time scanning. Integration of multi-axis operations can also be accomplished with the DB-8151 module as the HSL architecture is ready to use on a parallel multi-axis controller.

HSL is ideal for:

- Distributed solutions based on a multi-axis controller architecture
- Daughter board solutions making the motion system configuration more flexible and compact.
- Simpler wiring for remote I/O, including discrete I/O and analog I/O modules
- A vast number of discrete I/O points (hundreds or more)
- Real-time and fast scanning
- High speed data acquisition

The block diagram of the DB-8151 is as follows.

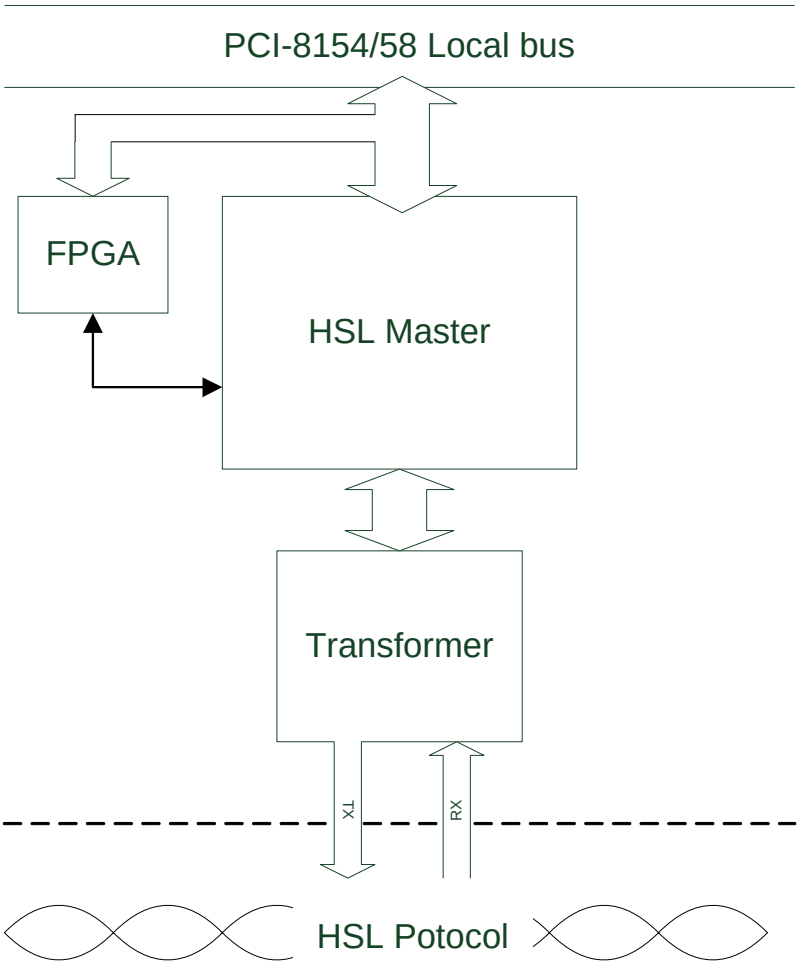


Figure 1-1: Block Diagram of the DB-8151

1.1 Specifications

Interface

- For use with the PCI-8154 and PCI-8158 PCI cards only
- The DB-8151-RJ45 must also be used when using the DB-8151

Master Controller

- HSL ASIC master control
- 48 MHz external clock

Interface

- RS-485 with transformer isolation
- Full duplex communication
- 3/6/12 Mbps transmission rate can be set by software (6 Mbps default)

Connector

- RJ45 connector x 2

Interrupt

- Status read back

LED Indicator

- Link status

Dimension

- 96.42 (L) x 62 (W) mm

Operating Temperature

- 0 to 60°C

Storage Temperature

- -20 to 80°C

Power Consumption

- +3.3V @ 250 mA (typical)
- +5V @100 mA (typical)

1.2 Supported Software

Program Library

ADLINK provides a Windows WDM driver and DLL function library for the DB-8151. These function libraries are shipped with the board and supports Windows 2000/XP.

2 Installation

This chapter describes how to install the DB-8151. Please follow these steps below:

- Check what you have (Section 2.1, page 7)
- Check the PCB (Section 2.2, page 8)
- Install the hardware (Section 2.3, page 10)
- Install the software driver (Section 2.4, page 12)
- Understanding the I/O signal connections (Chapter 3, page 17) and their operations (Chapter 4, page 43)

2.1 Package Contents

In addition to this User's Guide, the package also includes the following items:

- DB-8151: HSL Master control board x1
- DB-8151-RJ45: HSL interface board x1
- Copper Pillar x4
- Screws x8
- 10 Pin Flat Cable x1

If any of these items are missing or damaged, contact the dealer from whom you purchased the product. Save the shipping materials and carton to ship or store the product in the future.

2.2 DB-8151 and DB-8151-RJ45 Outline Drawing

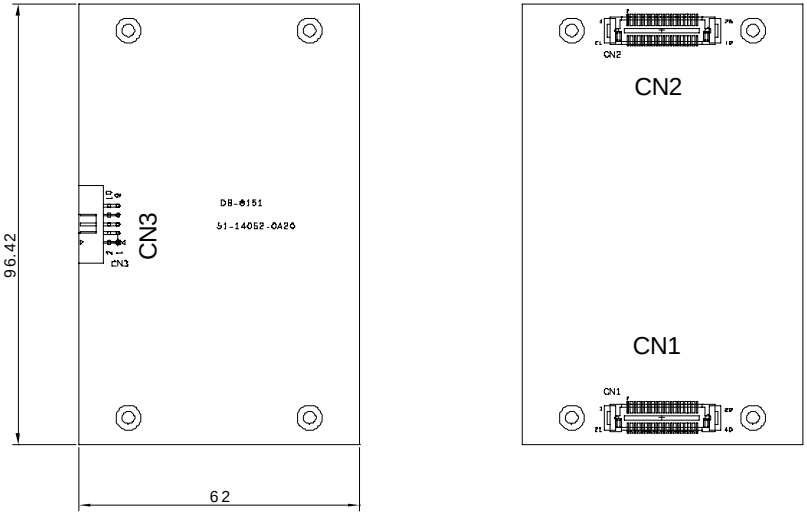


Figure 2-1: DB-8151 PCB Layout

DB-8151:

- CN1, CN2: Daughter board connectors for the PCI-8154/58
- CN3: Main signal connector

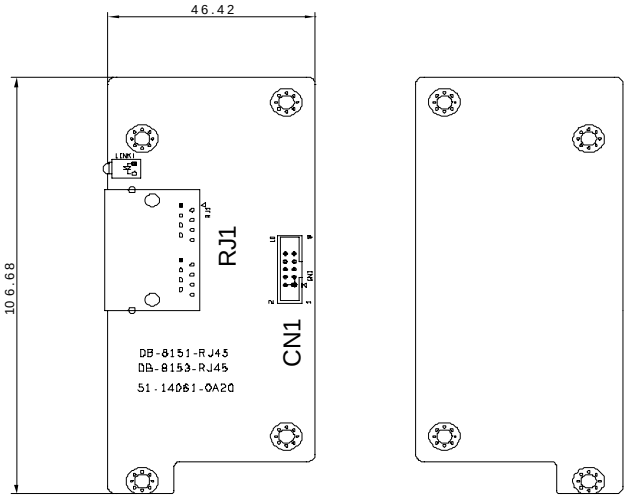


Figure 2-2: DB-8151-RJ45 PCB Layout

DB-8151-RJ45:

- RJ1: Main HSL master controller connector
- CN1: Link to DB-8151

2.3 DB-8151 Hardware Installation

2.3.1 Hardware configuration

DB-8151 must be installed on to a PCI-8154 or PCI-8158 only. Please ensure correct orientation of the DB-8151 before it attaching it to the PCI-8154 or PCI-8158.

2.3.2 Installation Procedures

Please follow installation procedure as follows.

Daughter Board Installation

1. Attach the DB-8151 on to the PCI-8154 or PCI-8158. Please ensure correct orientation of the DB-8151 daughter board.
2. Screw the eight screws into the corresponding copper pillars.
3. Connect the DB-8151 and DB-8151-RJ45 with the 10-pin flat cable.

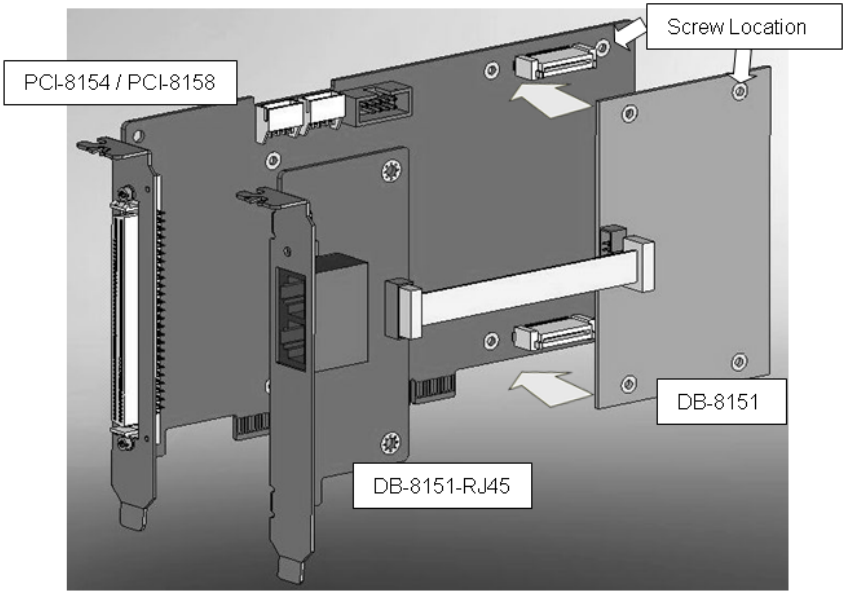


Figure 2-3: Board Configuration

Carrier Board Installation

4. Turn off the computer. Turn off all accessories (printer, modem, monitor, etc.) connected to the computer. Remove the cover from the computer.
5. Select one available 32-bit PCI expansion slot with a neighboring open slot for the DB-8151-RJ45. PCI slots are shorter than ISA or EISA slots and are usually white or ivory.
6. Before handling the PCI-8154 or PCI-8158 with DB-8151 and DB-8151-RJ45, discharge any static buildup on your body by touching the metal case of the computer. Hold the edge of the card and do not touch the components.
7. Plug the PCI-8154/58 with DB series vertically down into the PCI slot, and then secure the PCI-8154/58 bracket and DB-8151-RJ45 bracket onto rear panel.

2.3.3 Troubleshooting:

If the system doesn't boot or if any erratic behavior of the PCI board is experience, it is most likely caused by an interrupt conflict (possibly an incorrect ISA setup). The solution, once determined it is not a simple oversight, is to consult the BIOS documentation that comes with your system.

Check the control panel of the Windows system if the card is listed by the system. If not, check the PCI settings in the BIOS or use another PCI slot.

2.4 Software Driver Installation

Execute the following steps:

1. Auto-run the ADLINK All-In-One CD.
2. Follow the procedures of the installation wizard.
3. After setup installation has completed, reboot the system.

2.5 DB-8151 Pin Assignments

2.5.1 CN3 Pin Description

CN3 is the main DB-8151 connector linking to DB-8151-RJ45 via a 10-pin flat cable.

PIN NO	PIN OUT
PIN 1	+5V
PIN 2	FG
PIN 3	DG
PIN 4	LED Signal
PIN 5	RXD1
PIN 6	TXD
PIN 7	RXD2
PIN 8	TXE
PIN 9	DG
PIN 10	FG

Table 2-1: DB-8151 CN3 Pin Assignment

2.6 DB-8151-RJ45 Pin Assignments

2.6.1 CN1 Pin Description

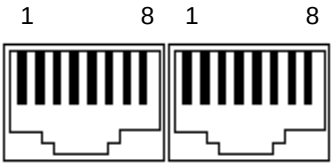
CN1 is the main DB-8151-RJ45 connector linking to the DB-8151 through a 10-pin flat cable.

PIN NO	PIN OUT
PIN 1	+5V
PIN 2	FG
PIN 3	DG
PIN 4	LED Signal
PIN 5	RXD1
PIN 6	TXD
PIN 7	RXD2
PIN 8	TXE
PIN 9	DG
PIN 10	FG

Table 2-2: DB-8151-RJ45 CN1 Pin Assignment

2.6.2 RJ1 Pin Description

RJ1 is the RJ45 connector on the DB-8151-RJ45.



PIN NO.	PIN OUT
PIN 1	NC
PIN 2	NC
PIN 3	RX+
PIN 4	TX-
PIN 5	TX+
PIN 6	RX-
PIN 7	NC
PIN 8	NC

Table 2-3: DB-8151-RJ45 RJ1 Pin Assignment

3 HSL Slave Module

The HSL is a master-slave network system that features an innovative distributed architecture that modularizes the communication, I/O functions and signal termination. ADLINK provides a complete line of slave I/O modules and terminal bases including discrete I/O, analog I/O, and motion control to meet your application requirements. For motion control modules, refer to the HSL-4XMO user's manual.

Slave I/O Module. There are three groups of slave I/O modules with varied dimensions. The slave I/O modules provide the terminal base with different levels of I/O capability. To identify each slave I/O module in an HSL network, an electronic data sheet is embedded in the module, and each module is identified by an address ID configurable via the 6-bit DIP switch. Depending on the I/O support, each slave I/O module may be assigned one or two address IDs. Since the highest ID number in an HSL master is 63 (6-bit and '0' reserved for master), up to 63 slave I/O modules are supported in one HSL master.

Terminal Base. Offers an easy wiring media. Both power and signal wiring go from the terminal base to the slave I/O modules. Also, master links to all slave I/O modules via the terminal base using an RJ-45 cable. The TB makes the slave I/O modules hot-swappable without interfering other modules in the same HSL network.

HUB/Repeater. Provides sub-system support for various network topologies.

Wiring Cable. The cables connecting the HSL master and slave I/O modules are standard 100 Base/TX with RJ-45 connectors. These are exactly the same with commercial Ethernet cables.

3.1 Slave I/O Module

3.1.1 Discrete I/O Module

ADLINK provides three I/O module series: DB, M and L.

- DB: Daughterboard form factor
- M: Daughterboard form factor with aluminum cover
- U: U-series

Series	Model	Discrete Input	Discrete Output	Relay Output	Slave Index Occupation
DB	HSL-DI32-DB-N/P	32			2 (Consecutive from odd number)
	HSL-DO32-DB-N/P		32		2 (Consecutive from odd number)
	HSL-DI16DO16-DB-N/P	16	16		1
M	HSL-DI32-M-N/P	32			2 (Consecutive from odd number)
	HSL-DO32-M-N/P		32		2 (Consecutive from odd number)
	HSL-DI16DO16-M-NN/NP/PN//PP	16	16		1
	HSL-R8DI16-M-N/P	16		8	1
U	HSL-DI16DO16-US/UJ-NN/NP/PN/PP	16	16		1
	HSL-DI16-UL	16			1

Below is the selection guide.

HSL	-	DIxDOx	-	x	-	XY

Discrete I/O Type:
DI16DO16: 16 discrete inputs and 16 discrete outputs
DI32: 32 discrete inputs
DO32: 32 discrete outputs
R8DI16: 8 relay outputs and 16 discrete inputs

Series:
DB: Daughter board form factor
M: Daughter board with aluminum cover
U: U-Series

Signal Type:
X: Input Signal Type: **N**PN sinking and **P**NP sourcing support
Y: Output Signal Type: **N**PN sinking and **P**NP sourcing support

3.1.2 Analog I/O Module

ADLINK provides M series analog I/O module.

Series	Model	Analog Input	Analog Output	Slave Index Occupation
M	HSL-AI16AO2-M-VV	16	2	2 (Leap number)
	HSL-AI16AO2-M-AV	16	2	2 (Leap number)
U	HSL-AO4		4	2

Below is the selection guide.

HSL	-	AIxAOx	-	x	-	XY
		Discrete I/O Type: AI16AO16: 16 analog inputs and 2 analog outputs		**Series**: M: Daughter board with aluminum cover		**Signal Type**: X: Input signal type, V means voltage and A means current. Y: Output signal type, V means voltage.

3.1.3 Motion Control

ADLINK provides the HSL-4XMO-CG-N/P and HSL-4XMO-CD-N/P remote motion control cards. Below is the selection guide.

HSL	-	4XMO	-	Cx	-	X
		Controllable Axes: 4XMO: 4-axis pulse train type motion control		**Series**: CG: The connection interface is general type. CD: The connection interface is D-sub 25.		**Signal Type**: X: Output Signal Type: **N**PN sinking and **P**NP sourcing support

The HSL-4XMO-CG-N/P is suitable for applications using stepper and linear motors. For HSL-4XMO-CD-N/P, ADLINK provides the accessories and transfer cable for direct connection to a servo amplifier. For details, refer to the HSL-4XMO user's manual.

3.1.4 General Specifications

Discrete I/O Module

Discrete Input	Photo couple isolation	2500 V_{RMS}	
	Input impedance	4.7 kΩ	
	Input Voltage	+24 V *	
	Input Current	For NPN(1)	-10 mA
		For PNP(2)	+10 mA
	Operation Voltage (@ 24 V_{DC} Power Supply)	For NPN(1)	ON: 11.4 V_{DC} (max) OFF: 14.3 V_{DC} (min)
		For PNP(2)	ON: 12.6 V_{DC} (min) OFF: 9.8 V_{DC} (max)
	Response Time	ON: 8.8 µs(Typical) ; OFF: 42 µs(Typical)	
Discrete Output	Switch capacity	For NPN(3)	All channels(5): -50mA/ch at 24V_{DC}
		For PNP(4)	All channels: +50mA/ch at 24V_{DC}
	Response Time	ON to OFF: 68 µs	
		OFF to ON: 1.1 µs	
Relay	Relay Type	SPST, normally open, non-latching	
	Rating	30 V_{DC}/2 A; 250 V_{AC}/2 A	
	Switching Frequency	20 times/minute at rated load	
	Response Time	ON to OFF: 3 µs (max)	
		OFF to ON: 6 µs (max)	

(1): NPN sinking type sensor input module

(2): PNP sourcing type sensor input modules

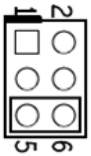
(3): NPN sinking type sensor output module

(4): PNP sourcing type sensor output modules

(5): U-series all channels: -90 mA at 24 V_{DC}

***Note**: The HSL-DI16-UL supports 5 V, 12 V, and 24 V, selected by a jumper for each channel:

- JDI0 - JDI15 (input voltage setting)



5 V
12 V
24 V (default)

Discrete Input	Input impedance	4.7 kΩ (@24 V_{DC}, 2.74 k (@12 V_{DC}), 1.1 k (@5 V_{DC})	
	Operation Voltage	DI_COM @ 24 V_{DC}	ON: 14.0 V_{DC} (max) OFF: 18.0 V_{DC} (min)
		DI_COM @ 12 V_{DC}	ON: 6.0 V_{DC} (min) OFF: 8.0 V_{DC} (max)
		DI_COM @ 5 V_{DC}	ON: 1.0 V_{DC} (min) OFF: 3.0 V_{DC} (max)

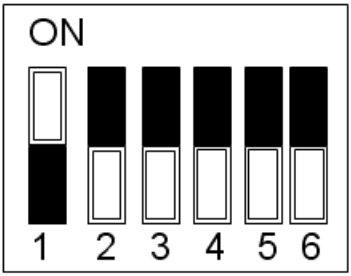
Analog I/O Module

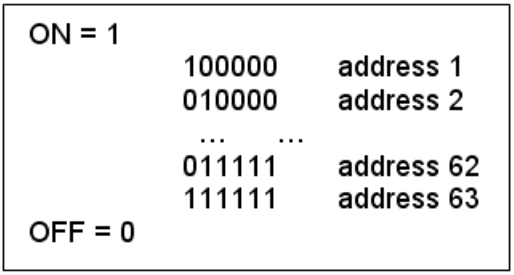
Analog Input	A/D Resolution	16-bit (14-bit guaranteed)
	Input Range	For VV type: ±10, ±5, ±2.5, ±1.25 V
		For AV type: 20 mA, 10 mA, 5 mA
	A/D Conversion	10 µs
	Signal Type	16-CH single-ended; 8-CH differential
Analog Output	D/A Resolution	16-bit
	DA Settling Time	10 µs

Motion Control

Refer to the HSL-4XMO user's manual.

3.1.5 DIP Switch Setting:





Notes

(1) The address (or slave index) 0 is reserved.

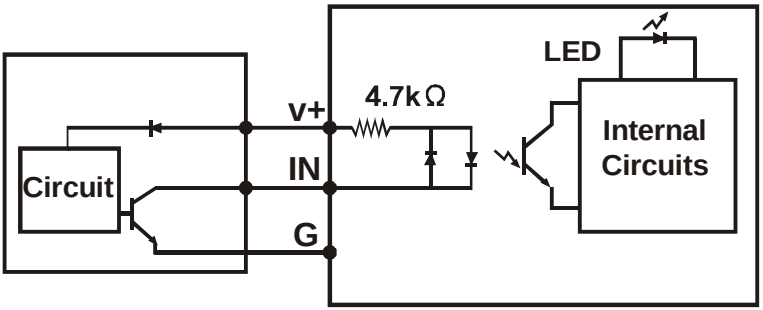
(2) The HSL-DI32-M, HSL-DO32-M, HSL-DI32-DB, and HSL-DO32-DB require two consecutive addresses starting from an odd number. For example, if the DIP switch is set to 3, it occupies slave index 3 and 4.

(3) The HSL-AI16AO2-M-VV/AV requires two leap addresses at full duplex mode. For example, if the DIP switch is set to 2, the module occupies addresses 2 and 4.

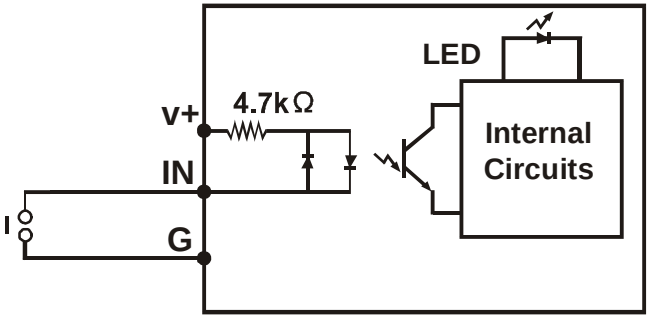
(4) The HSL-4XMO-CG-N/P and HSL-4XMO-CD-N/P require four leap addresses at full duplex mode. For example, if the DIP switch is set to 2, these modules occupy 2, 4, 6, and 8. At half duplex mode, it requires 4 consecutive addresses.

3.1.6 Wiring Diagram

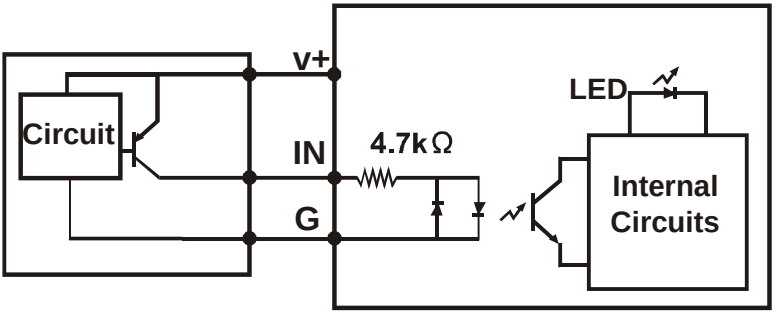
-N NPN Sinking type sensor Input



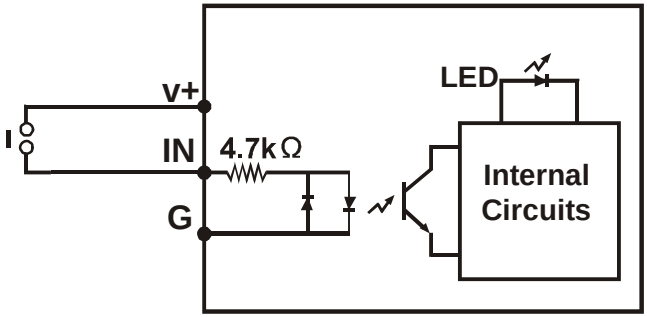
-N Dry Contact Input



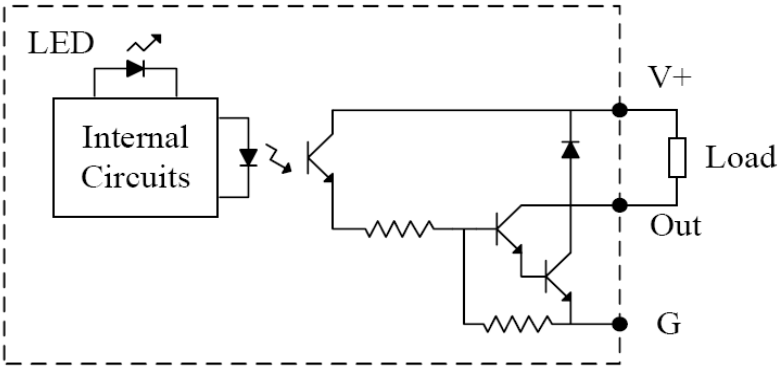
-P PNP Sourcing type sensor Input



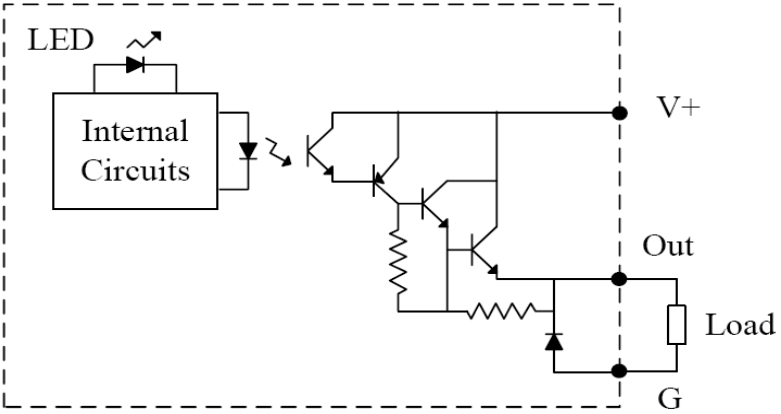
-P Wet Contact Input



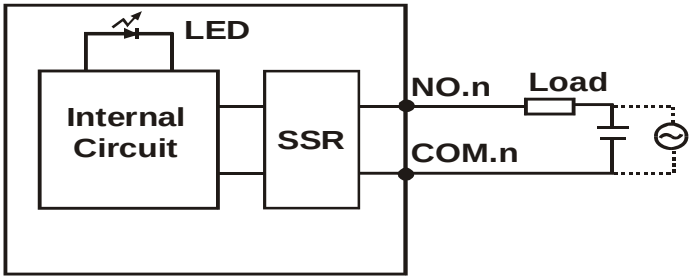
-N NPN Sinking Output



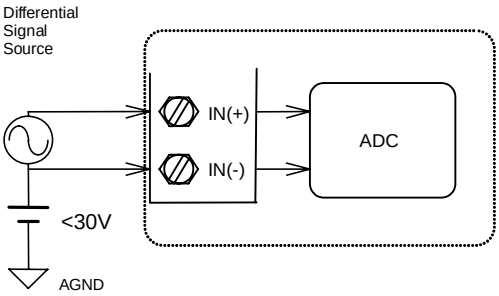
-P PNP Sourcing Output



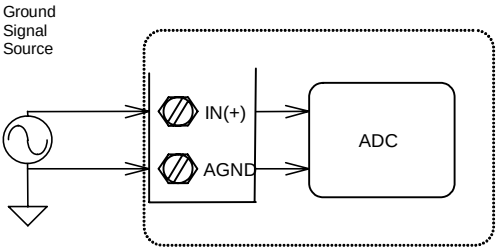
-R Relay Output



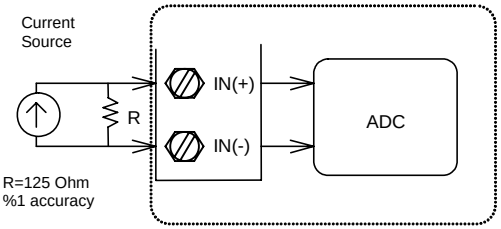
Analog Input (Differential Voltage Input)



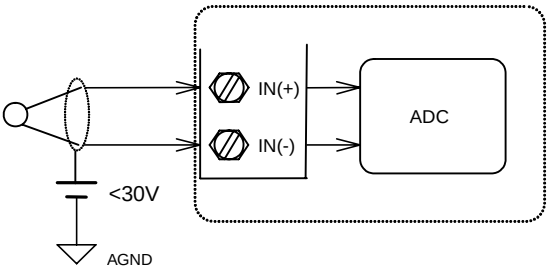
Analog Input (Single-End Voltage Input)



Analog Input (Current Measure)

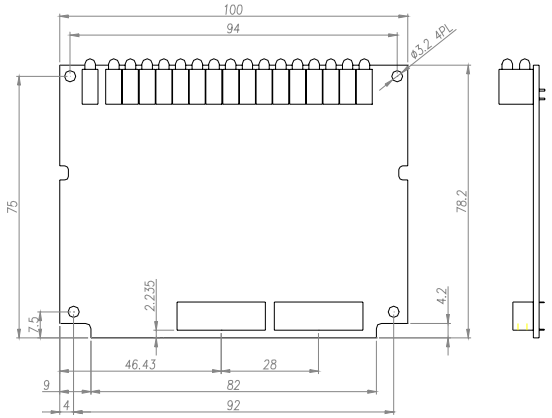


Thermocouple Measurement

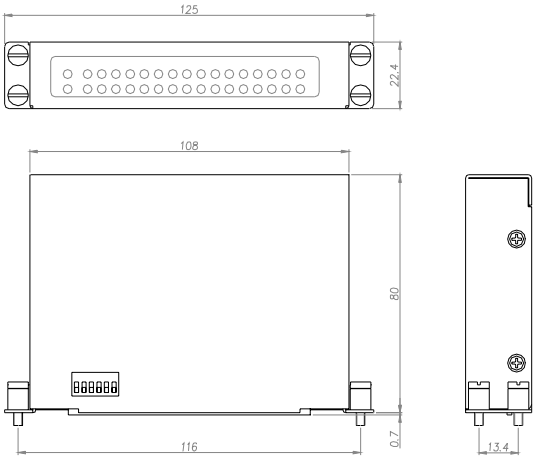


Dimension

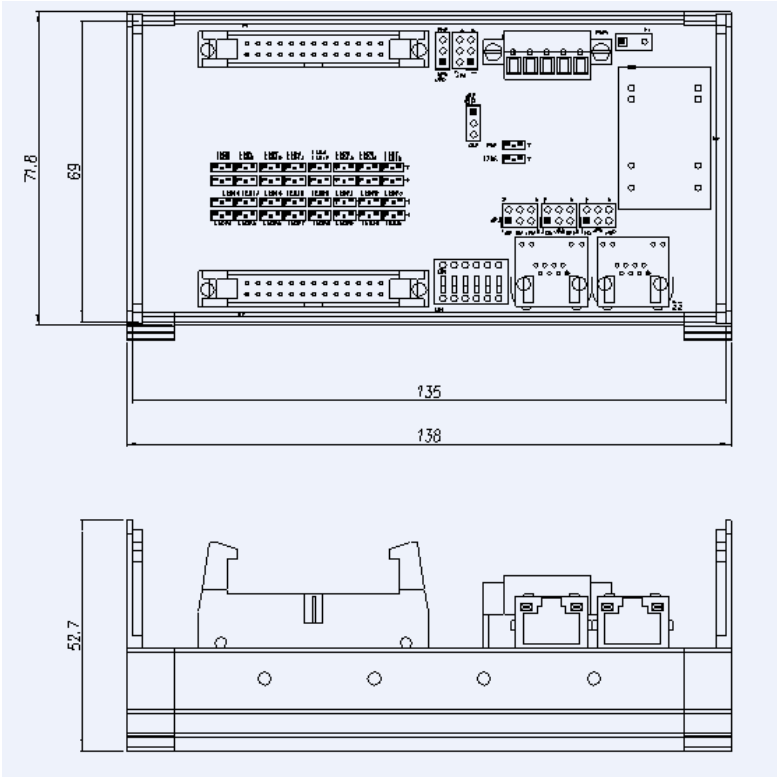
-DB Daughterboard form factor (100 mm X 78.2 mm)



-M Daughterboard with aluminum cover (125 mm X 80 mm)



-U U-series slave I/O module (71.8 mm X 138 mm)



3.2 Terminal Base

Available terminal bases include:

- HSL-TB32-U-DIN
- HSL-TB64-DIN
- HSL-TB32-M-DIN
- HSL-TB32-MD

Features

- Field I/O wiring connection for HSL I/O modules
- Screw- or spring-type terminal for easy field wiring
- Power and ground connections for each signal channel
- Interlocking design for installation in rugged environments
- Power LED indicator
- DIN rail mounting
- Onboard terminator resistor

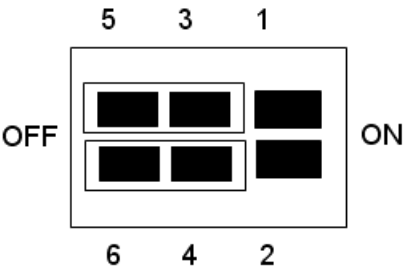
3.2.1 General Description

	Model Name	Specifications	Supports
For DB Series	HSL-TB32-U	(1): 32 channels direct connected terminal base (2): One DB slot	All HSL DB-series modules
	HSL-TB64	(1): 64 channels direct connected terminal base (2): Two DB slots	All HSL DB-series modules
For M Series	HSL-TB32-M HSL-TB32-MD	32 channels direct connected terminal base for HSL M-series module	All HSL M-series modules

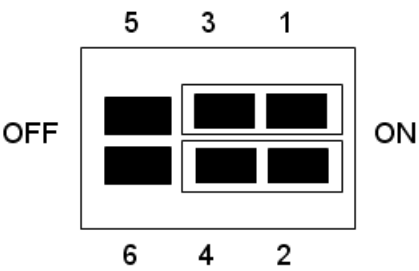
3.2.2 Jumper Settings

Since HSL is a serial transmission system, a terminator must be placed at the end of the cable. Each TB has an onboard jumper selectable terminator. **The terminator must be enable only by the last module**.

Not the last module (Default)

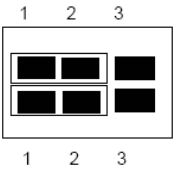


The last module



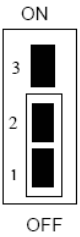
3.2.3 HSL-TB32-MD Jumper Settings

JP1,2 (External Power Option)



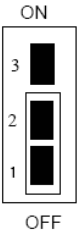
1, 2 short: Different Power (Default)
2, 3 short: Common Power

JP3 (Tx Terminal Resistor)



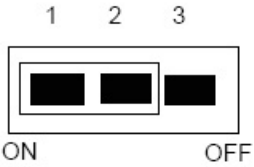
OFF is default setting

JP4 (Rx Terminal Resistor)



OFF is default setting

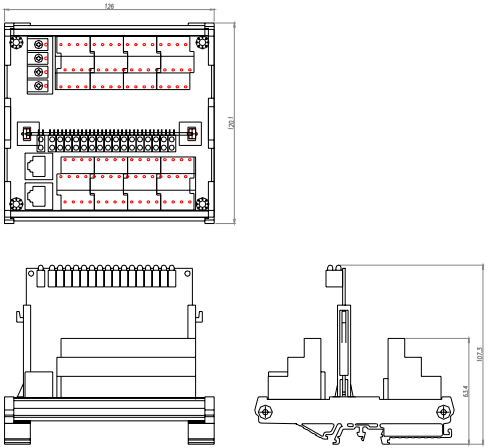
JP5 (Fuse Option)



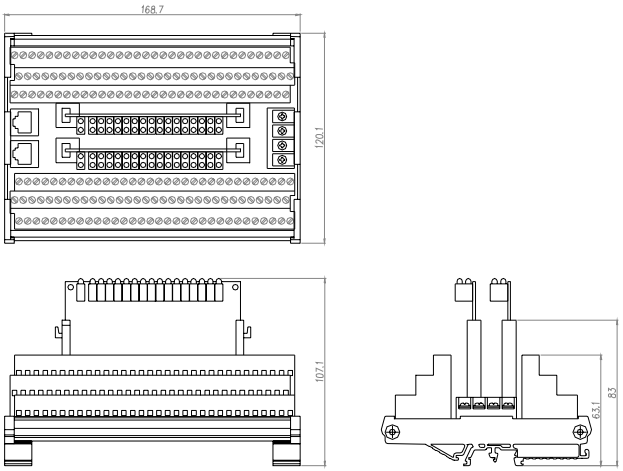
1, 2 short: With Fuse
2, 3 short: Without Fuse (Default)

3.2.4 Dimensions

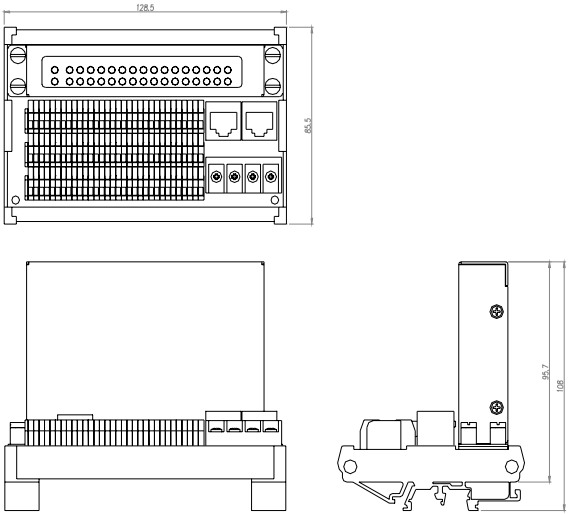
-DB with HSL-TB32-U-DIN (126 mm x 120.1 mm x 107.3 mm)



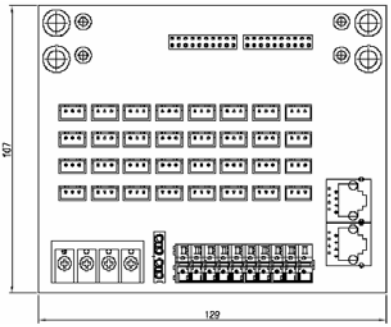
-DB with HSL-TB64-DIN (168.7 mm x 120.1 mm x 107.3 mm)



-M module with HSL-TB32-M-DIN (128.5 mm x 85.5 mm x 108 mm)



-HSL-TB32-MD (129 mm x 107 mm)



3.3 HSL-HUB/Repeater

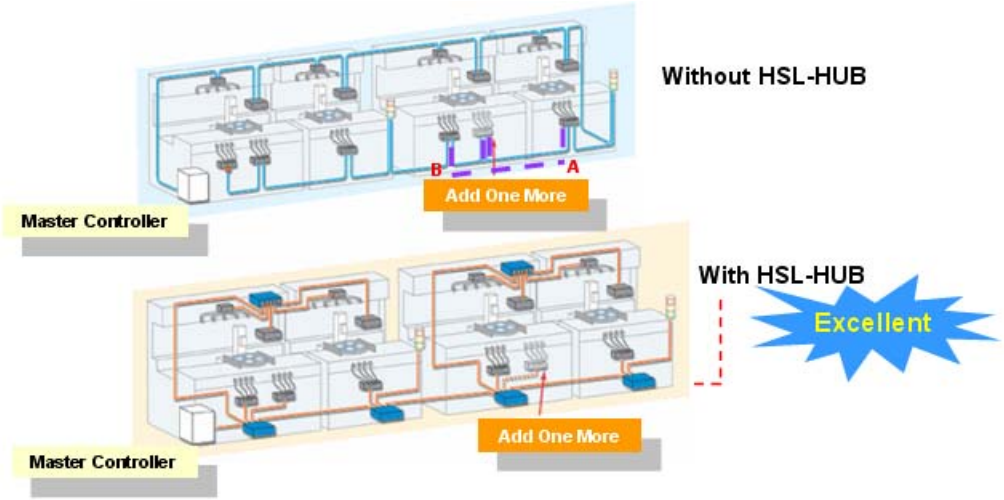
Available HSL-HUB/Repeater includes:

- HSL-HUB
- HSL-Repeater

Features

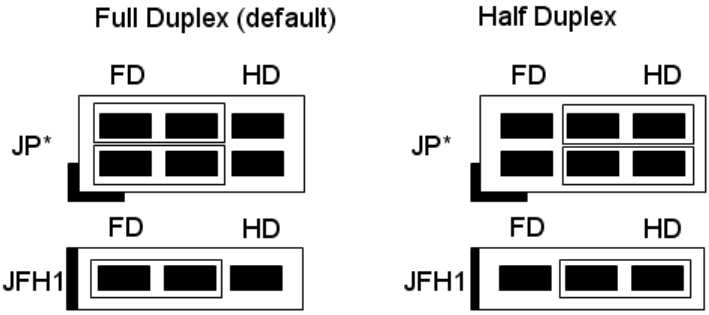
- Master to HUB, HUB to HUB, HUB to Slave link styles
- Supports T-bracing connection/Star connection (subsystem concept)
- Supports up to 2.4 km wiring distance via seven HSL-HUB/ Repeater modules
- One input port with three output segment ports
- Jumper configurable 3/6/12 Mbps transmission speed
- Jumper configurable full and half duplex transmission modes
- RJ-45 phone jack for easy installation
- 24 VDC input

3.3.1 General Description

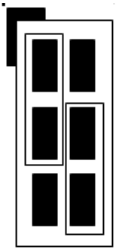


3.3.2 Jumper Setting

FD/HD setting JP*(0 to 3), JFH1



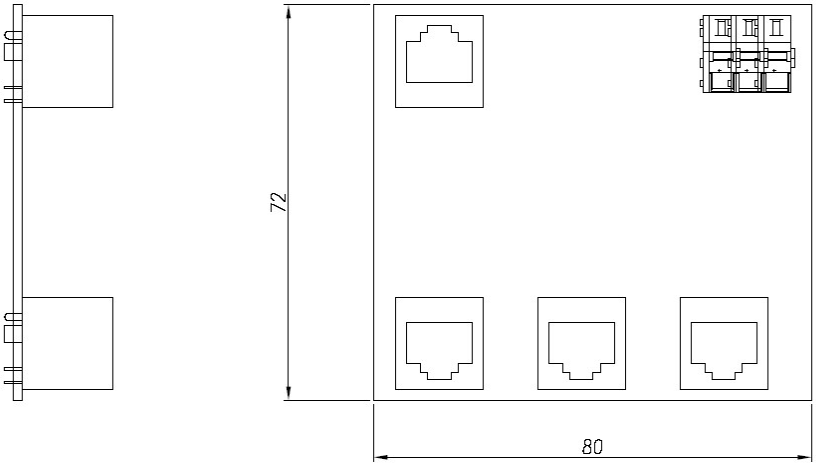
3/6/12 Mbps setting: JBPS1



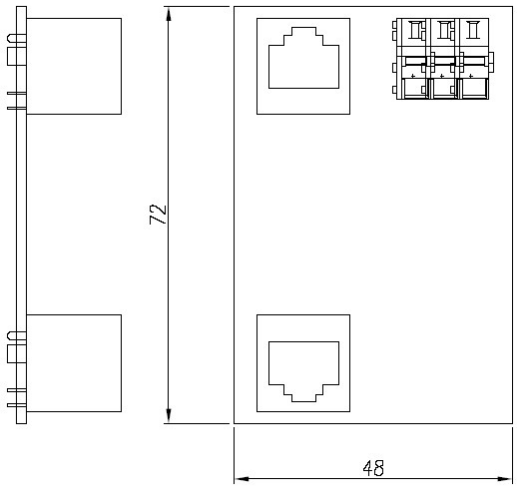
1 – 3 and 2 – 4	12 Mbps
1 – 3 and 4 – 6	6 Mbps (default)
3 – 5 and 2 – 4	3 Mbps
3 – 5 and 4 – 6	EXC

3.3.3 Dimensions

HSL-HUB



HSL-Repeater



3.4 Managing Slave Index in an HSL Network

3.4.1 Before you proceed

Before powering on the slave modules, you have to adjust the DIP switch. For more information, refer to section 3.1.6. Take note of the following:

1. A master controller can connect up to 63 slave indexes. For example, the PCI-7852 has two master controllers. Therefore, it supports a maximum 126 slave indexes.
2. The more compact the slave addresses are in an HSL network, the greater efficiency.
3. Observe the discrete I/O and relay module rule.

Module	Slave Index Occupation	Transmission Mode	Transmission Speed
HSL-DI16DO16-M-NN/NP/PN/PP	1 (Any address)	Full Duplex (Fixed)	6 Mbps (Fixed)
HSL-DI16DO16-DB-NN/NP/PN/PP			
HSL-R8DI16-M-N/P			
HSL-DI8-L-N/P			
HSL-DO8-L-N/P			
HSL-DI4DO4-L-NN/NP/PN/PP			
HSL-DI16-UL			
HSL-DI16DO16-UJ/US			
HSL-DI32-M-N/P	2 (Consecutive from odd number)		
HSL-DI32-DB-N/P			
HSL-DO32-M-N/P			
HSL-DO32-DB-N/P			

4. Observe the analog I/O and thermocouple module rule.

Module	Slave Index Occupation	Transmission Mode	Transmission Speed
HSL-AI16AO2-M-VV	2 (Leap number)	Full Duplex (Fixed)	3/6/12 Mbps Selectable
HSL-AI16AO2-M-AV			
HSL-AO4-U			

Observe the motion control module rule

Module	Slave Index Occupation	Transmission Mode	Transmission Speed
HSL-4XMO-CG-N/P	4 (Leap) / 4(Consecutive)	Full Duplex / Half Duplex	3/6/12 Mbps Selectable
HSL-4XMO-CD-N/P			

5. Special rules

   ▷ If you will install only one HSL-AI16AO2-M-VV or HSL-AI16AO2-M-AV and the DIP switch is set to 1 (HSL-AI16AO2-M-VV/AV only supports full duplex mode), the occupied indexes will be 1 and 3. You must assign the parameter “max_slave_No” of HSL_start(…) as 4 to ensure correct communication. You may ignore this rule when using the HSL_auto_start function.

   ▷ If you will install only one HSL-4XMO-CG-N/P or HSL-4XMO-CD-N/P and the DIP switch is set as to 1 and full duplex mode, the occupied indexes will be 1, 3, 5, and 7. You must assign the parameter “max_slave_No” of HSL_start(…) as 8 to ensure correct communication. You may ignore this rule when using the HSL_auto_start function. In half duplex mode, these modules occupy 1, 2, 3 and 4. Therefore, you must assign the “max_slave_No” of HSL_start(…) as 4 or call the HSL_auto_start function.

3.4.2 Examples

The following examples are provided for user reference. All modules used are set in full duplex mode.

Example 1

Provided you installed two HSL-DI16DO16, two HSL-DI32-M-N, and an HSL-AI16AO2-VV with all slave modules in full duplex mode, you can have two conditions as follows:

Condition 1: HSL-AI16AO2-VV operates at 6 Mbps.

We recommended that you use the provided slave index configuration.

Item	DIP Switch	Index Occupation in HSL
HSL-DI32-M-N #1	1	1, 2
HSL-DI32-M-N #2	3	3, 4
HSL-AI16AO2-VV	5	5, 7
HSL-DI16DO16 #1	6	6
HSL-DI16DO16 #2	8	8

This is an example of a compact composition. The scan time needs 30.33 µs x 8 at 6 Mbps, full duplex mode. Users can connect the modules with one master controller.

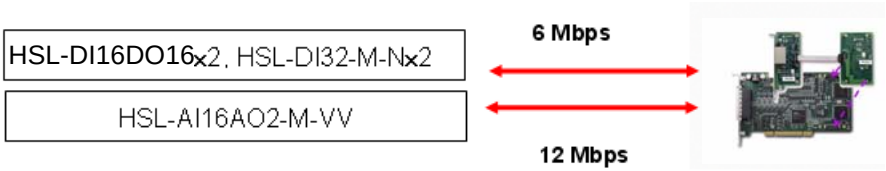
Condition 2: HSL-AI16AO2-VV operates at 12 Mbps.

We recommended that you use the provided slave index configuration.

Item	DIP Switch	Index Occupation in HSL
HSL-DI32-M-N #1	1	1, 2
HSL-DI32-M-N #2	3	3, 4
HSL-DI16DO16 #1	5	5
HSL-DI16DO16 #2	6	6

Another example of a compact composition. The scan time needs 30.33 µs x 6 at 6 Mbps, full duplex mode. You may connect these modules with one master controller. The HSL-AI16AO2-M-VV module connects to another master controller. The DIP switch of

HSL-AI6AO2-M-VV is assigned as 1. Because of the special rule, users have to assign the “max_slave_No” of HSL_start(…) as 3 or call HSL_auto_start by connect_index #1. The illustration below explains this.



Consequently, the cycle time of the first master controller is 30.33 µs x 6 and the cycle time of the second master controller is 45.5 µs at 12 Mbps, full duplex mode.

Example 2

Provided you installed two HSL-DI16DO16-UJ, one HSL-DI16DO16-M-NN, two HSL-DO32-M-N, one HSL-AI16AO2-VV, and two HSL-4XMO-CG-N with all slave modules in full duplex mode, you can have the following conditions:

Condition 1: The HSL-AI16AO2-VV and two HSL-4XMO-CG-N operate in 6 Mbps.

We recommended that you use the provided slave index configuration.

Item	DIP Switch	Index Occupation in HSL
HSL-4XMO-CG-N #1	1	1, 3, 5, 7
HSL-4XMO-CG-N #2	2	2, 4, 6, 8
HSL-DO32-M-N #1	9	9, 10
HSL-DO32-M-N #2	11	11, 12
HSL-AI16AO2M-VV	13	13, 15
HSL-DI16DO16-UJ #1	14	14
HSL-DI16DO16-UJ #2	16	16
HSL-DI16DO16-M-NN	17	17

The scan time needs 30.33µ x 17 at 6 Mbps, full duplex mode. You can connect these modules with one master controller.

Condition 2: An HSL-AI16AO2-VV and two HSL-4XMO-CG-N modules operate at 12 Mbps.

We recommended that you use the provided slave index configuration.

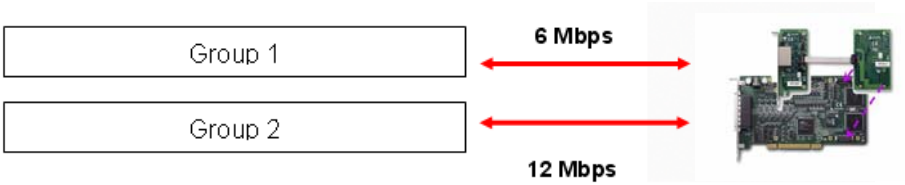
Group 1	DIP Switch	Index Occupation in HSL
HSL-DO32-M-N #1	1	1, 2
HSL-DO32-M-N #2	3	3, 4
HSL-DI16DO16-UJ #1	5	5
HSL-DI16DO16-UJ #2	6	6
HSL-DI16DO16-M-NN	7	7

The scan time needs 30.33µs x 7. You may connect these modules with one master controller. The HSL-AI16AO2-M-VV

and two HSL-4XMO-CG-N modules connect to another master controller. The management table below is for reference.

Group 2	DIP Switch	Index Occupation in HSL
HSL-4XMO-CG-N #1	1	1, 3, 5, 7
HSL-4XMO-CG-N #2	2	2, 4, 6, 8
HSL-AI16AO2-M-VV	9	9, 11

Refer to the illustration below.



The cycle time of the first master controller is 30.33µs x 7, while the cycle time of second master controller is 15.17µs x 11 at 12 Mbps, full duplex mode.

4 HSL LinkMaster Utility for DB-8151

After installing the HSL master controller and slave modules, install the PCI-8154/58 driver and use the LinkMaster utility to test and debug the system. This utility provides a user-friendly interface to easily test I/O status, including read/write the I/O data, calibration, and motion control. It is strongly recommended that this utility be used before implementing the whole system.

Note: LinkMaster is only available for Windows 2k/XP with screen resolution higher than 800×600.

4.1 Software Installation

You can install the HSL drivers from the ADLINK All-in-One CD that comes with the package or you may download the drivers from the ADLINK website. The latest driver version are available from the website.

1. Find the SETUP.exe and execute the file.
2. The follwoing screen will appear.

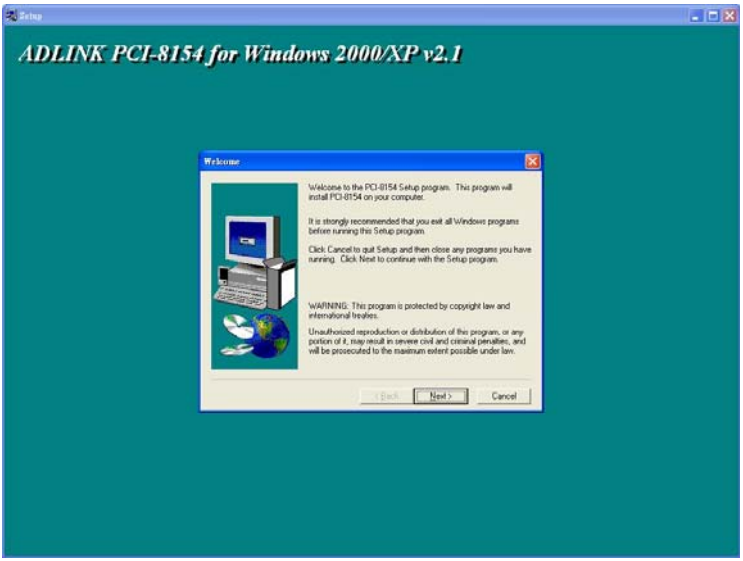


Figure 4-1: PCI-8154/58 Installation Page

Follow the instruction to complete installation.

3. After complete the installation processes, you have to restart Windows system so that the PCI-8154/58 drivers could work normally.

4.2 ADLINK HSL LinkMaster Utility

4.2.1 Launching the LinkMaster Utility

After installing the drivers, click Start > PCI-8154 > LinkMaster_8154_DB8151 to launch the LinkMaster utility. The main window appears.

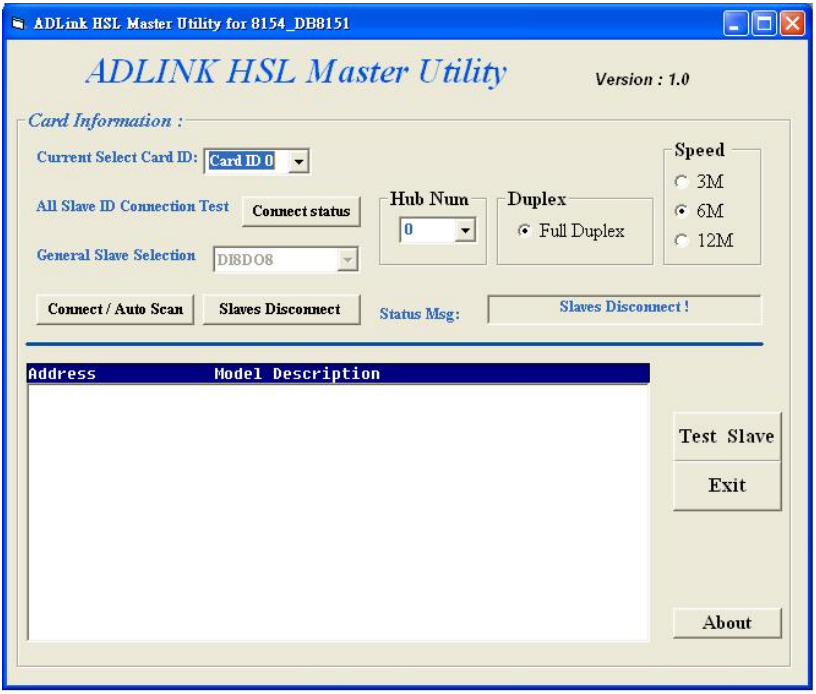


Figure 4-2: HSL LinkMaster Utility

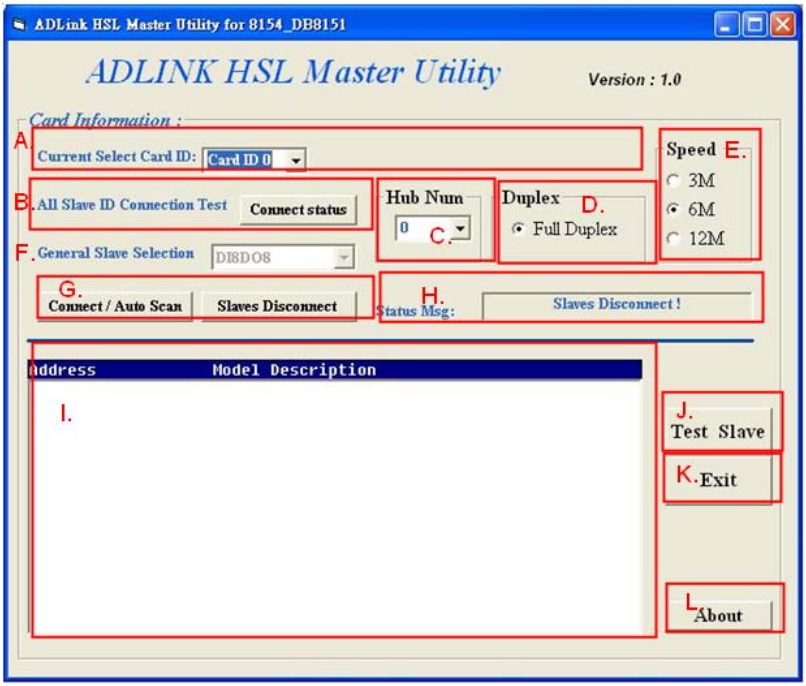
4.2.2 About the LinkMaster Utility

Before running the LinkMaster utility, please review the following notes should be reviewed.

1. LinkMaster is a testing and debugging program based on VB 6.0 and is only available for Windows® 98/NT/2000/XP environments with a monitor that has a screen resolution of 800x600 or higher. The utility does not support DOS environment.
2. The LinkMaster version control may be found on the top-right corner of the main window.
3. Any slave modules may be tested with this utility, including discrete I/O, analog I/O, thermocouple module, and motion control modules. For motion control utility and manipulation, refer to the HSL-4XMO user's manual.

4.2.3 LinkMaster Utility Introduction

Below is the LinkMaster main user interface labeled according to function.

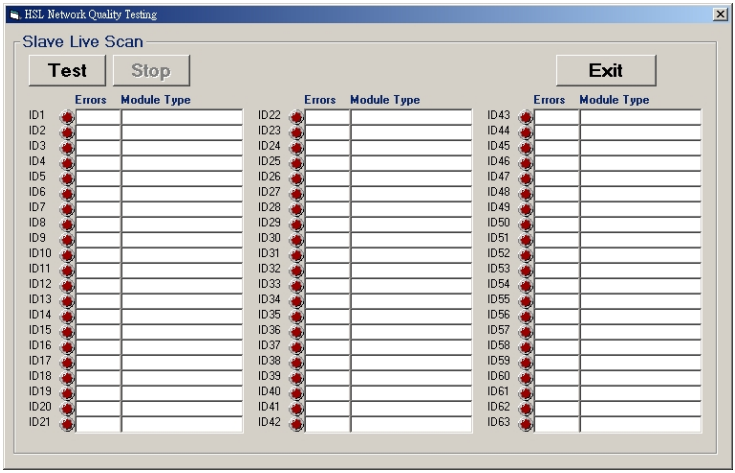


- A. Select card
- B. Network quality test
- C. Set hub number
- D. Set duplex mode
- E. Set speed mode
- F. General slave selection
- G. Auto scan slave modules
- H. Show software information
- I. Show module information
- J. Test slave module
- K. Exit LinkMaster
- L. Version information

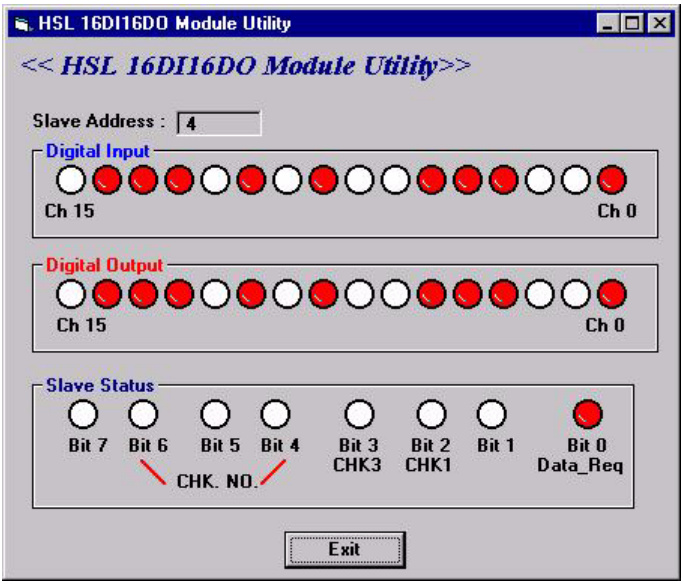
Below are descriptions of the main interface buttons.

1. **Current Select Card ID**. When LinkMaster is activated, it searches all HSL master control cards installed in the system. Every card shows its index (ID) starting from 0. This function can be used to specify which card to operate.
2. **Current Select Connect Index**. For 8154_DB8151, only CN0 is available..
3. **ALL Slave ID Connection Test**. The screen capture below shows a live scan of all I/O modules for network quality test. The LinkMaster lets you check the network environment.

   Start the test by clicking on the **Test** button. Press **Stop** to stop scanning. When you start the test, the utility continuously tests each ID and shows the module type to left-column labels. Right-column labels show the counter for communication error.



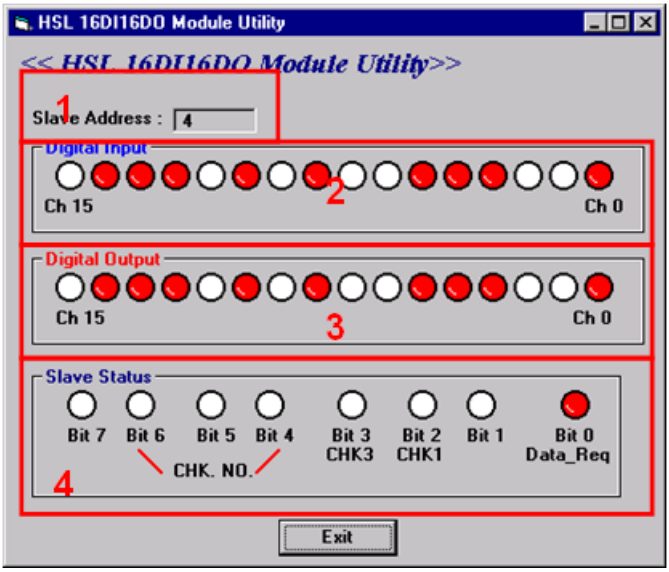
4. **Connect/Auto Scan**. Clicking this button allows the utility to scan all slave modules connected to the master card with specified connect index. The utility shows all the slave modules' information including the address and slave type within the 9th block.
5. **Slaves Disconnect**. Click this to stop the utility from scanning all the slave modules and to disconnect them.
6. **Status Msg**. Checks if the slave modules are connected or disconnected.
7. **Test Slave**: While all connected slave modules list in 9th block, you can use this function to activate the testing dialog. For example, when you connect the HSL-DI16DO16-M-NN, you will see this module from the screen. Clicking on it will show a window from where you can test and debug the modules.



8. **Exit**. Click to close the utility.
9. **About**. Shows the DLL version information.

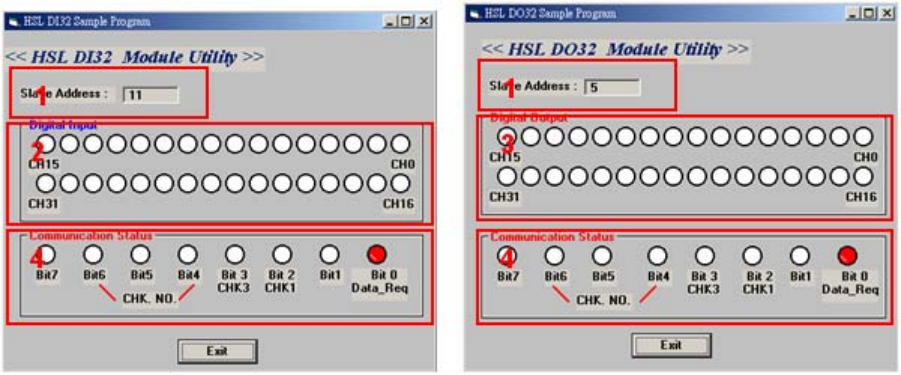
The succeeding sections outline the usage of the slave module utility.

4.2.4 HSL-DI16DO16 Utility



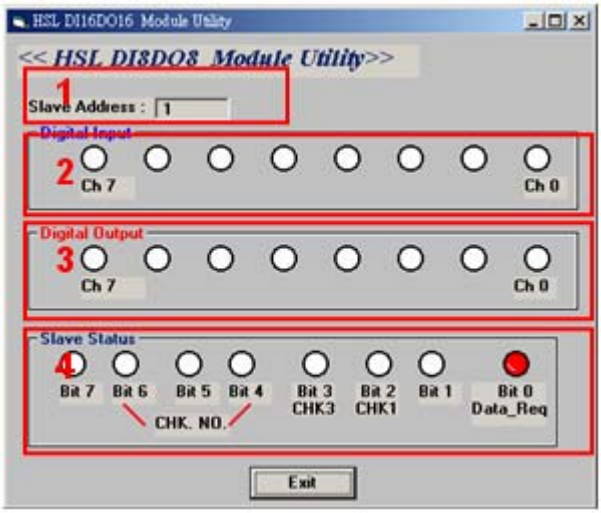
1. **Slave Address**. Shows the slave index occupied by the module.
2. **Digital Input**. A white circle indicates no digital input; a red icon indicates that the digital input is not activated.
3. **Digital Output**. Click on the icon to activate the digital output. Red icon indicates that the digital output is turned on, and vice-versa.
4. **Slave Status**: Shows the communication status between the slave module and the master card. The functions definition are enumerated below.
   - ▷ Bit 0 is Data_Req bit.
   - ▷ Bit 2 is for CHK1. When Bit 2 is equal to 1, a communication error occurred once.
   - ▷ Bit 3 is for CHK3. When Bit 3 is equal to 1, a communication error occurred three times.
   - ▷ Bit 4, Bit 5 and Bit 6 bits are for CHK7. When Bit 4, Bit 5, and Bit 6 are all equal to 1, a communication error occurred seven times.

4.2.5 HSL-DI32 and HSL-DO32 Utility



1. **Slave Address**. Shows the slave index occupied by the module. These modules occupy two slave indexes starting from an odd number. For example, when you adjust the DIP switch to 3, the modules are assigned indexes 3 and 5.
2. **Digital Input**. A white circle indicates no digital input; a red icon indicates that the digital input is not activated.
3. **Digital Output**. Click on the icon to activate the digital output. Red icon indicates that the digital output is turned on, and vice-versa.
4. **Slave Status**: Shows the communication status between the slave module and the master card. The functions definition are enumerated below.
   - ▷ Bit 0 is Data_Req bit.
   - ▷ Bit 2 is for CHK1. When Bit2 is equal to 1, a communication error occurred once).
   - ▷ Bit 3 is for CHK3. When Bit3 is equal to 1, a communication error occurred three times.
   - ▷ Bit 4, Bit 5 and Bit 6 bits are for CHK7. When Bit 4, Bit 5, and Bit 6 are all equal to 1, a communication error occurred seven times.

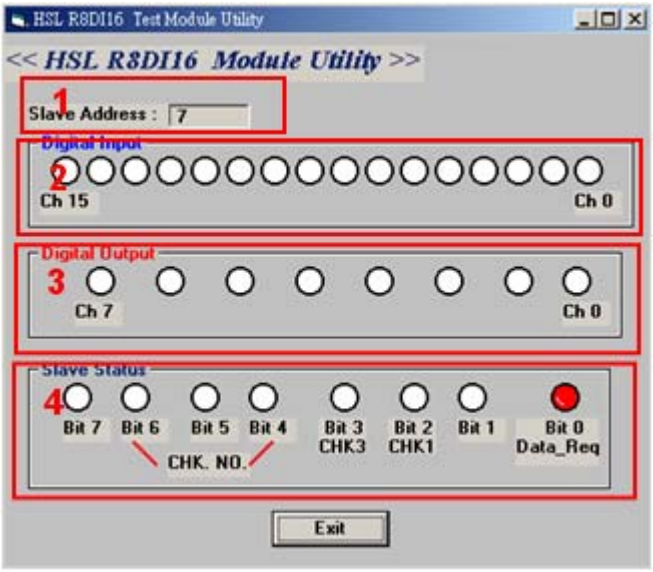
4.2.6 HSL-DI8/HSL-DO8/HSL-DI4DO4 Utility



The HSL-DI8-L, HSL-DO8-L, and HSL-DI4DO4-L series all have the same utility dialog as above. All contents are described as follows.

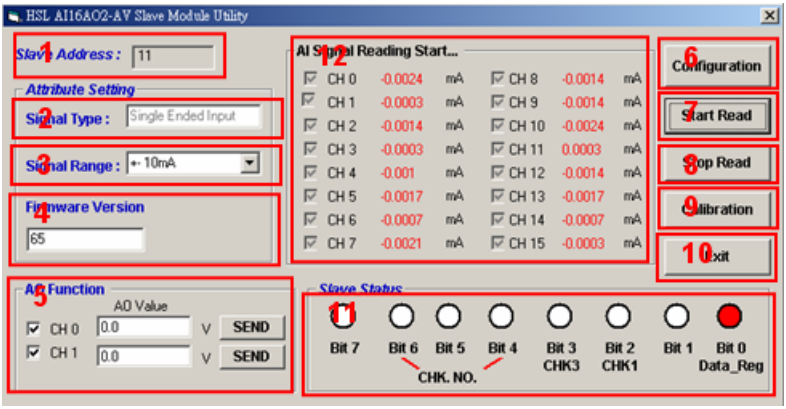
1. **Slave Address**. Shows the slave index occupied by the module. These modules occupy only one slave index.
2. **Digital Input**. A white circle indicates no digital input; a red icon indicates that the digital input is not activated.
3. **Digital Output**. Click on the icon to activate the digital output. Red icon indicates that the digital output is turned on, and vice-versa.
4. **Slave Status**: Shows the communication status between the slave module and the master card. The functions definition are enumerated below.
   - ▷ Bit 0 is Data_Req bit.
   - ▷ Bit 2 is for CHK1. When Bit2 is equal to 1, a communication error occurred once.
   - ▷ Bit 3 is for CHK3. When Bit3 is equal to 1, a communication error occurred three times.
   - ▷ Bit 4, Bit 5 and Bit 6 bits are for CHK7. When Bit 4, Bit 5, and Bit 6 are all equal to 1, a communication error occurred seven times.

4.2.7 HSL-R8DI16 Utility



1. **Slave Address**. Shows the slave index occupied by the module. These modules occupy only one slave index.
2. **Digital Input**. A white circle indicates no digital input; a red icon indicates that the digital input is not activated.
3. **Digital Output**. Click the icon to activate digital output. This function turns the relay ON or OFF. A red circle indicates that the relay is on, and vice versa.
4. **Slave Status**: Shows the communication status between the slave module and the master card. The functions definition are enumerated below.
   - ▷ Bit 0 is Data_Req bit.
   - ▷ Bit 2 is for CHK1. When Bit 2 is equal to 1, a communication error occurred once.
   - ▷ Bit 3 is for CHK3. When Bit 3 is equal to 1, a communication error occurred three times.
   - ▷ Bit 4, Bit 5 and Bit 6 bits are for CHK7. When Bit 4, Bit 5, and Bit 6 are all equal to 1, a communication error occurred seven times.

4.2.8 HSL-AI16AO2 Utility



1. **Slave Address**. Shows the slave index occupied by the module. The module occupies two consecutive indexes. For example, when you adjust the DIP switch to 4, the module obtains slave indexes 4 and 6.
2. **Signal Type**. Indicates the module's signal type.
3. **Signal Range**. Allows selection of the signal range. The utility offers four ranges including ±10 V, ±5 V, ±2.5 V and ±1.25 V for HSL-AI16AO2-M-VV. For HSL-AI16AO2-M-AV, the signal ranges are 20 mA, 10 mA, and 5 mA.
4. **Firmware Version**. Shows the latest firmware version.
5. **AO Function**. Key in the analog output value in the text box, then press SEND to trigger the AO. The range is ±10 V.
6. **Configuration**. Allows you to check if the signal range is correct before clicking on the Start Read button. The Configuration button allows you to save the information and complete the configuration task.
7. **Start Read**. Enables the A/D conversion task to read back the analog input values. The values are shown in the 12th block.
8. **Stop Read**. Disables the A/D conversion task.

9. **Calibration**. Calibrates the module. The modules are shipped with correct calibration. Refer to Appendix C if you want to recalibrate the module.
10. **Exit**. Closes the utility.
11. **Slave Status**: Shows the communication status between the slave module and the master card. The functions definition are enumerated below.
    - ▷ Bit 0 is Data_Req bit.
    - ▷ Bit 2 is for CHK1. When Bit 2 is equal to 1, a communication error occurred once).
    - ▷ Bit 3 is for CHK3. When Bit 3 is equal to 1, a communication error occurred three times.
    - ▷ Bit 4, Bit 5 and Bit 6 bits are for CHK7. When Bit 4, Bit 5, and Bit 6 are all equal to 1, a communication error occurred seven times.

4.2.9 HSL-4XMO Utility

Refer to the HSL-4XMO user's manual.

5 Function Library: PCI-8154 + DB-8151

This chapter describes the function library for the DB-8151 HSL system. These functions can be used to develop programs in C, C++, or Visual Basic.

5.1 List of Functions

This section details all the functions. The function prototypes and some common data types are decelerated in **PCI_8154.h**. We suggest you to use these data types in your application programs. The following table shows the data type names and their range.

Type Name	Description	Range
U8	8-bit ASCII character	0 to 255
I16	16-bit signed integer	-32768 to 32767
U16	16-bit unsigned integer	0 to 65535
I32	32-bit signed long integer	-2147483648 to 2147483647
U32	32-bit unsigned long integer	0 to 4294967295
F32	32-bit single-precision floating-point	-3.402823E38 to 3.402823E38
F64	64-bit double-precision floating-point	-1.797683134862315E308 to 1.797683134862315E309
Boolean	Boolean logic value	TRUE, FALSE

All function calls have the same **_8154_db51_HSL_** prefix when using the PCI-8154 as the carrier. If the PCI-8158 is selected as the primary motion controller, all functions will carry the **_8158_db51_HSL_** prefix. The **_8154_db51_HSL_** prefix is used below for use with the PCI-8154. If functions belong to a system level purpose, the convention will be as follows.

_8154_db51_HSL_{action_name}.

e.g. **_8154_db51_HSL_initial**().

If they belong to discrete I/O modules purpose, the function will be as follows.

_8154_db51_HSL_D_{action_name}.

e.g. **_8154_db51_HSL_D_read_input**()

If they belong to analog I/O modules purpose, the function will be as follows.

_8154_db51_HSL_A_{action_name}.

e.g. **_8154_db51_HSL_A_write_output**().

If they belong to motion control modules purpose, the function will be as follows.

_8154_db51_HSL_M_{action_name}.

e.g. **_8154_db51_HSL_M_start_tr_move**().

As for the motion control library description, please refer to HSL-4XMO function library manual. This manual only contains the system level function, discrete I/O control, and analog I/O control.

Initialization & System Information, Section 5.2

Function Name	Description
_8154_db51_HSL_initial	DB8151 Master card initialization
_8154_db51_HSL_close	Release all resources occupied by DB8151 master card
_8154_db51_HSL_start	Start to scan all the salve modules connected to DB8151 master card
_8154_db51_HSL_auto_start	Start to scan and automatically detect all the slave modules connected to DB8151 master card
_8154_db51_HSL_stop	Stop scanning the connected slave modules
_8154_db51_HSL_set_scan_condition	Set scanning conditions
_8154_db51_HSL_get_scan_condition	Get scanning conditions
_8154_db51_HSL_connect_status	Get the communication status of the specified slave module
_8154_db51_HSL_slave_live	Get the module status of the slave module

Discrete I/O, Section 5.3

Function Name	Description
_8154_db51_HSL_D_read_input	Read back all discrete I/O with unsigned 32-bit
_8154_db51_HSL_D_read_channel_input	Read back discrete I/O by channel selection
_8154_db51_HSL_D_write_output	Write all discrete I/O with unsigned 32-bit
_8154_db51_HSL_D_write_channel_output	Write discrete I/O by channel selection
_8154_db51_HSL_D_read_ouput	Read back the output value stored in RAM
_8154_db51_HSL_D_read_all_slave_input	Read back all inputs of slave modules
_8154_db51_HSL_D_write_all_slave_output	Write all outputs of slave modules
_8154_db51_HSL_D_set_input_logic	Set the logic of digital input
_8154_db51_HSL_D_set_output_logic	Set the logic of digital output
_8154_db51_HSL_D_set_int_renewal_type	Set DI renewal check type
_8154_db51_HSL_D_set_int_renewal_bit	Set the data bits of DI renewal check for each slave
_8154_db51_HSL_D_set_int_control	Set DI interrupt enable or disable
_8154_db51_HSL_D_wait_di_interrupt	Wait DI renewal event

Analog I/O, Section 5.4

Function Name	Description
_8154_db51_HSL_A_start_read	Start A/D conversion.
_8154_db51_HSL_A_stop_read	Stop A/D conversion
_8154_db51_HSL_A_set_signal_range	Set the signal range of analog input channels
_8154_db51_HSL_A_get_signal_range	Get the signal range of analog input channels
_8154_db51_HSL_A_get_input_mode	Get the signal input mode
_8154_db51_HSL_A_set_last_channel	Set the last channel of analog input channels
_8154_db51_HSL_A_get_last_channel	Get the last channel of analog input channels
_8154_db51_HSL_A_read_input	Read back the value of analog input channels
_8154_db51_HSL_A_write_output	Send out the analog output
_8154_db51_HSL_A_read_output	Read back the analog output data
_8154_db51_HSL_A_sync_rw	Read and write the data synchro-nously
_8154_db51_HSL_A_get_version	Get the kernel version of analog I/O module

5.2 Initialization & System Information

@ Name

_8154_db51_HSL_initial – Master board initialization

_8154_db51_HSL_close – Release all resource occupied by master board

_8154_db51_HSL_start – Start to scan all slave module connected to master board

_8154_db51_HSL_auto_start – Start to scan and automatically detect all the slave modules connected to master card

_8154_db51_HSL_stop – Stop scanning the connected slave modules

_8154_db51_HSL_set_scan_condition – Set scanning conditions

_8154_db51_HSL_get_scan_condition – Get scanning conditions

_8154_db51_HSL_connect_status – Get the communication status of the specified slave module

_8154_db51_HSL_slave_live – Get the module status of the slave module

@ Description

_8154_db51_HSL_initial:

Initialize the hardware and software states of the 8154_DB8151 HSL master card. Before calling this function, users must call _8154_initial first to get resource of DB8151. Users can check the return code of this function to know if the initialization is successful or not.

_8154_db51_HSL_close:

This function is to release the resource occupied by the 8154_DB8151 HSL master card. When terminating the program, do not forget to call this function to release all the resource occupied by 8154_DB8151 HSL card.

_8154_db51_HSL_start:

This function is used to scan the total connected slave modules. Users can assign how many slave indexes HSL master board should scan. Then, it will scan from 1 to the specified value.

_8154_db51_HSL_auto_start:

This function is used to automatically detect the total connected slave modules. Every master controller can connect up to 63 slave indexes. As a result, It will scan from 1 to 63.

_8154_db51_HSL_stop:

This function is used to stop scanning the connected slave modules.

_8154_db51_HSL_set_scan_condition:

This function is used to assign the scan rate(3/6/12M) and communication types(Full/half duplex). And this function needs to set up between the function **_8154_db51_HSL_initial** and **_8154_db51_HSL_start**.

_8154_db51_HSL_get_scan_condition:

By this function, User can get the settings of communication types and scan rate which are set by "**_8154_db51_HSL_set_scan_condition**".

_8154_db51_HSL_connect_status:

This function is used to check the communication status between master board and slave modules.

_8154_db51_HSL_slave_live:

This function is used to check the status of the slave module (live or die).

@ Syntax

C/C++ (DOS, Windows 98/NT/2K/XP)

```
I16 _8154_db51_HSL_initial (U16 card_ID);
I16 _8154_db51_HSL_close (U16 card_ID);
I16 _8154_db51_HSL_start (U16 card_ID, U16
     max_slave_No);
I16 _8154_db51_HSL_auto_start (U16 card_ID);
I16 _8154_db51_HSL_stop (U16 card_ID);
I16 _8154_db51_HSL_set_scan_condition(I16
     card_ID, I16 transfer_rate, I16 hub_number);
I16 _8154_db51_HSL_get_scan_condition(I16
     card_ID, I16 *transfer_rate, I16
     *hub_number);
I16 _8154_db51_HSL_connect_status (U16 card_ID,
     U16 slave_No, U8 *sts_data);
I16 _8154_db51_HSL_slave_live (U16 card_ID, U16
     slave_No, U8 *live_data);
```

Visual Basic (Windows 98/NT/2K/XP)

```
B_8154_db51_HSL_initial (ByVal card_ID As
     Integer) As Integer
B_8154_db51_HSL_close (ByVal card_ID As Integer)
     As Integer
```

```
B_8154_db51_HSL_start (ByVal card_ID As Integer,
     ByVal max_slave_No As Integer) As Integer
B_8154_db51_HSL_auto_start (ByVal card_ID As
     Integer) As Integer
B_8154_db51_HSL_stop (ByVal card_ID As Integer)
     As Integer
B_8154_db51_HSL_set_scan_condition(ByVal card_ID
     As Integer, ByVal transfer_rate As Integer,
     ByVal hub_number As Integer);
B_8154_db51_HSL_get_scan_condition((ByVal
     card_ID As Integer, transfer_rate As
     Integer, hub_number As Integer);
B_8154_db51_HSL_connect_status (ByVal card_ID As
     Integer, ByVal slave_No As Integer, sts_data
     as Byte) As Integer
B_8154_db51_HSL_slave_live (ByVal card_ID As
     Integer, ByVal slave_No as Integer,
     live_data as Byte) As Integer
```

@ Argument

`card_ID`: Specify the HSL master card index. Normally, the board index sequence would be decided by the system. The index is from 0.

`max_slave_No`: The maximum slave index connected to the HSL master card with the connect_index. The valid value is from 1 to 63.

`slave_No`: Specifiy the slave module with slave index which want to perform this function. The valid value is from 1 to 63.

`transfer_rate`: transfer rate setting

- 1: 3M
- 2: 6M
- 3: 12M

`hub_number`: Cascaded Hub number. If no Hub in the system, the value of hub_number is set to 0.

***sts_data**: The communication status of this slave module. The definition is as follows.

- Bit 0 is Data_Req bit.
- Bit 2 is for CHK1. (If Bit2 is 1. It means that there is 1 time communication error).
- Bit 3 is for CHK3. (If Bit3 is 1. It means that there are 3 times communication errors).
- Bit 4, BIT 5 and BIT 6 bits are for CHK7. (If Bit4, Bit5 and Bit6 all are 1. It means that there are 7 times communication errors).

***live_data**: The module status.

- 1: the module is live
- 0: the module is die.

@ Return Code

```
ERR_No_Error
ERR_Open_Driver_Fail
ERR_Invalid_Board_Number
ERR_Satellite_Number
ERR_Connect_Index
```

5.3 Discrete I/O

@ Name

_8154_db51_HSL_D_read_input – Read back all discrete I/O with unsigned

32-bit

_8154_db51_HSL_D_read_channel_input – Read back discrete I/O by channel selection

_8154_db51_HSL_D_write_output – Write all discrete I/O with unsigned 32-bit _8154_db51_HSL_D_write_channel_output –Write discrete I/O by channel selection

_8154_db51_HSL_D_read_ouput – Read back the output value stored in RAM

_8154_db51_HSL_D_read_all_slave_input – Read back all inputs of slave modules

_8154_db51_HSL_D_write_all_slave_output – Write all outputs of slave modules

_8154_db51_HSL_D_set_input_logic – Set the logic of digital input

_8154_db51_HSL_D_set_output_logic – Set the logic of digital output

_8154_db51_HSL_D_set_int_renewal_type – Set DI renewal check type

_8154_db51_HSL_D_set_int_renewal_bit – Set the data bits of DI renewal check for each DI slave module

_8154_db51_HSL_D_set_int_control – Set DI interrupt enable or disable

_8154_db51_HSL_D_wait_di_interrupt – Wait DI renewal event

@ Description

`_8154_db51_HSL_D_read_input`:

This function is to read the digital input value of the discrete I/O module. Users have to specify the connect index and slave index.

`_8154_db51_HSL_D_read_channel_input`:

This function is to read the digital input value of the discrete I/O module at the specified channel.

`_8154_db51_HSL_D_write_output`:

This function is to write the digital output value of the discrete I/O module. Users have to specify the connect index and slave index.

`_8154_db51_HSL_D_write_channel_output`:

This function is to write the digital output value of the discrete I/O module at the specified channel.

`_8154_db51_HSL_D_read_ouput`:

This function is to write all digital output values to all connected discrete I/O modules. It will map all data into memory. With this function, user can write all digital output values to the all connected discrete I/O modules at one time.

`_8154_db51_HSL_D_read_all_slave_input`:

This function is used to read the digital input values from all slave I/O modules. In this function, all digital input values from all slave I/O modules can be read at one time.

_8154_db51_HSL_D_write_all_slave_output:

This function is to write the digital output values from all slave I/O modules. In this function, all digital output values from all slave I/O modules can be written at one time.

_8154_db51_HSL_D_set_input_logic:

This function is to set the digital input logic to the specified slave I/O module. The slave I/O module‘s address is slave_No.

_8154_db51_HSL_D_set_output_logic:

This function is to set the digital output logic to the specified slave I/O module. The slave I/O module‘s address is slave_No.

_8154_db51_HSL_D_set_int_renewal_type:

This function is used to set the types of hardward interrupt occurrence timing. There are 3 kind of types that are described as follow.

Type 1: Generates hardware interrupt when any DI datas transition are detected.

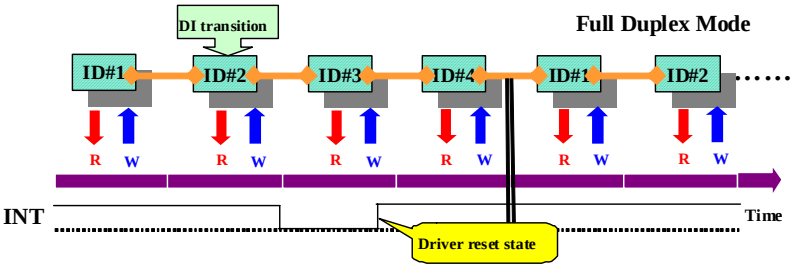


Figure 5-1: HSL DI Interrupt for Example#1

Type 2: Generates hardware interrupt when any DI datas transition are detected and when the scan cycle is completed.

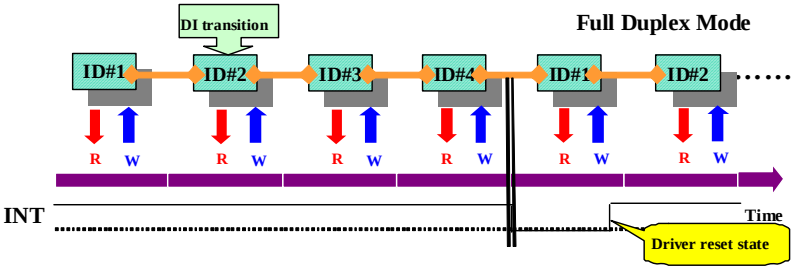


Figure 5-2: HSL DI Interrupt for Example#2

Type 3: Generates hardware interrupt when any DI datas transition are detected and when the scan cycle is completed.When interrupt occurred the scan will paused until driver reset the state.

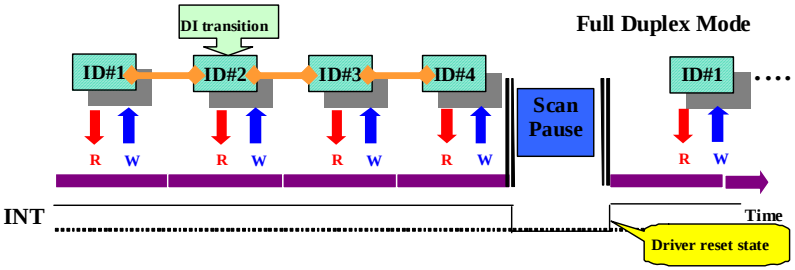


Figure 5-3: HSL DI Interrupt for Example#3

Caution: Scanning is paused while user choice the type3 of renewal type. This pause time depends on the user system performance. Consequently, when using type3, constancy(always keeping scan cycle constant) will not be maintained between scans.

_8154_db51_HSL_D_set_int_renewal_bit:

This function is used to set the Di data bits of specified modules that user want to monitor the states.

_8154_db51_HSL_D_set_int_control:

This function is used to enable or disabel the DI interrupt.

_8154_db51_HSL_D_wait_di_interrupt:

When user enabled the Interrupt function by **_8154_db51_HSL_D_set_int_control**() and set the renewal type and data bits on specified slave DI modules by **_8154_db51_HSL_D_set_int_renewal_bit**(),**_8154_db51_HSL_D_set_int_renewal_type**(). This function can be used to wait on a specific interrupt. When this function is running, the process would never stop until a even is triggered or the function is time out. The following code is an example.

```
I16 ret;
_8154_db51_HSL_D_set_int_renewal_type(1, 1);
 // slave id = 1, monitor the states of bit 0 and
     bit 1
_8154_db51_HSL_D_set_int_renewal_bit(1, 1,
     0x003);
_8154_db51_HSL_D_set_int_control(1, 1); //enable
…
// start wait
ret =_8154_db51_HSL_D_wait_di_interrupt(1,
     10000);
if(ret == ERR_No_Error)
{    // DI state trainisted and check which bits
     change states…
}else
{    // time out
}…
```

@ Syntax

C/C++ (DOS, Windows 98/NT/2K/XP)

```
I16 _8154_db51_HSL_D_write_output (I16 card_ID,
     I16 slave_No, U32 out_data);
I16 _8154_db51_HSL_D_write_channel_output(I16
     card_ID, I16 slave_No, I16 channel, U16
     out_data);
I16 _8154_db51_HSL_D_read_input (I16 card_ID, I16
     slave_No, U32 *in_data);
I16 _8154_db51_HSL_D_read_channel_input (I16
     card_ID, I16 slave_No, I16 channel, U16
     *in_data);
I16 _8154_db51_HSL_D_read_output (I16 card_ID,
     I16 slave_No, U32 *out_data_in_ram);
I16 _8154_db51_HSL_D_read_all_slave_input (I16
     card_ID, U16 *in_data);
I16 _8154_db51_HSL_D_write_all_slave_output (I16
     card_ID, U16 *out_data);
I16 _8154_db51_HSL_D_set_input_logic (I16
     card_ID, I16 slave_No, I16 input_logic);
I16 _8154_db51_HSL_D_set_output_logic (I16
     card_ID, I16 slave_No, I16 output_logic);
I16 _8154_db51_HSL_D_set_int_renewal_type(I16
     card_ID, I16 type);
I16 _8154_db51_HSL_D_set_int_renewal_bit(I16
     card_ID, I16 slave_No, U16 bitsOfCheck);
I16 _8154_db51_HSL_D_set_int_control(I16 card_ID,
     I16 enable);
I16 _8154_db51_HSL_D_wait_di_interrupt(I16
     card_ID, I32 time_out_ms);
```

Visual Basic (Windows 98/NT/2K/XP)

```
B_8154_db51_HSL_D_write_output (ByVal card_ID As
      Integer, ByVal slave_No As Integer, ByVal
      out_data As Long) As Integer
B_8154_db51_HSL_D_write_channel_output (ByVal
      card_ID As Integer, ByVal slave_No As
      Integer, ByVal channel As Integer, ByVal
      out_data As Integer) As Integer
HSL_D_read_input (ByVal card_ID As Integer, ByVal
      slave_No As Integer, in_data As Long) As
      Integer
B_8154_db51_HSL_D_read_channel_input (ByVal
      card_ID As Integer, ByVal slave_No As
      Integer, ByVal channel As Integer, in_data
      As Integer) As Integer
B_8154_db51_HSL_D_read_output (ByVal card_ID As
      Integer, ByVal slave_No As Integer,
      out_data_in_ram As Long) As Integer
B_8154_db51_HSL_D_read_all_slave_input (ByVal
      card_ID As Integer, in_data As Integer) As
      Integer
B_8154_db51_HSL_D_write_all_slave_output (ByVal
      card_ID As Integer, out_data As Integer) As
      Integer
B_8154_db51_HSL_D_set_input_logic (ByVal card_ID
      As Integer, ByVal slave_No As Integer, ByVal
      input_logic As Integer) As Integer
B_8154_db51_HSL_D_set_output_logic (ByVal card_ID
      As Integer, ByVal slave_No As Integer, ByVal
      output_logic As Integer) As Integer
B_8154_db51_HSL_D_set_int_renewal_type(ByVal
      card_ID As Integer, ByVal type As Integer)As
      Integer
B_8154_db51_HSL_D_set_int_renewal_bit(ByVal
      card_ID As Integer, ByVal slave_No As
      Integer, ByVal bitsOfCheck As Long) As
      Integer
```

@ Argument

card_ID: Specify the HSL master card index. Normally, the board index sequence would be decided by the system. The index is from 0.

slave_No: Specifies the slave module with slave index which want to perform this function. The valid value is from 1 to 63.

out_data: The digital output of this discrete module. The definition is as follows.

- For **HSL_D_write_output**: The data of channel 0 is assigned to bit 0; the data of channel 1 is assigned to bit 1 and so on.
- For **HSL_D_write_channel_output**: The value is digital output data of the specified channel.

***out_data**: It is a unsigned short array pointer. User have to create an unsigned short array which contain 63 cells. The cell index is corresponding to slave index. For example, cell index 0 corresponds to the module which slave index is 1. The cell index 1 corresponds to the module which slave index is 2 and so on. The last cell index 62 corresponds to the module which slave index is 63.

Cell index of array (Unsigned short)	Corresponding slave index
0	1
1	2
.....	.....
62	63

***in_data**: The input data of slave modules. The definition is as follows.

- For **HSL_D_read_input**: The data of channel 0 is assigned to bit 0; the data of channel 1 is assigned to bit 1 and so on.
- For **HSL_D_read_channel_input**: The value is digital input data of the specified channel.
- For **HSL_D_all_slave_index**: It is a unsigned short array pointer. User have to create an unsigned short array which contain 63 cells. The cell index is corresponding to

slave index. For example, cell index 0 corresponds to the module which slave index is 1. The cell index 1 corresponds to the module which slave index is 2 and so on. The last cell index 62 correpsonds to the module which slave index is 63.

Cell index of array (Unsigned short)	Corresponding slave index
0	1
1	2
…...	……
62	63

channel: Specify the channel of the discrete I/O module which want to perform this function. The valid value is described as follows.

- HSL-R8DI16: 0 - 15
- HSL-DI16DO16: 0 - 15
- HSL-DI32: 0 - 31
- HSL-DO32: 0 - 31

***out_data_in_ram**: The output data stored in RAM. The data of channel 0 is assigned to bit 0; the data of channel 1 is assigned to bit 1 and so on.

input_logic: Set the input logic to the specified module.

output_logic: Set the output logic to the specified module.

Type: Type of hardward interrupt occurrence timing. Value is 1-3

bitsOfCheck: renewal data bits(16bits)

enable: enable (0) or disable (1) the Di interrupt

time_out_ms: Specifies the time-out interval, in milliseconds. The function returns if the interval elapses, even if the interrupt is nonsignaled. If time_out_ms is zero, the function tests the Di state and returns immediately. If time_out_ms is -1, the function's time-out interval never elapses(infinite).

@ Return Code

```
ERR_No_Error
ERR_Invalid_Board_Number
ERR_Memory_Mapping
ERR_Connect_Index
ERR_Satellite_Number
ERR_Over_Max_Address
ERR_DI_Renewal_Type
ERR_Wait_Di_Interrupt
ERR_Di_Event_Open_Already
ERR_Di_Event_Disable
```

5.4 Analog I/O

@ Name

_8154_db51_HSL_A_start_read – Start A/D conversion

_8154_db51_HSL_A_stop_read – Stop A/D conversion

_8154_db51_HSL_A_set_signal_range – Set the signal range of analog input channels

_8154_db51_HSL_A_get_signal_range – Get the signal range of analog input channels

_8154_db51_HSL_A_get_input_mode – Get the signal input mode

_8154_db51_HSL_A_set_last_channel – Set the last channel of analog input channels

_8154_db51_HSL_A_get_last_channel – Get the last channel of analog input channels

_8154_db51_HSL_A_read_input – Read back the value of analog input channels

_8154_db51_HSL_A_write_output – Send out the analog output

_8154_db51_HSL_A_read_output – Read back the analog output data

_8154_db51_HSL_A_sync_rw – Read and write the data synchronously

_8154_db51_HSL_A_get_version – Get the kernel version of analog I/O module

@ Description

_8154_db51_HSL_A_start_read:

This function is used to initialize the reading operation of the analog input channels of all HSL AI/O modules which are connected to the master card.

Before using **_8154_db51_HSL_A_read_input**(), **_8154_db51_HSL_A_write_output**() and **_8154_db51_HSL_A_sync_rw**(), the functions need to be executed first to start the A/D conversion.

_8154_db51_HSL_A_stop_read:

This function is used to stop the reading operation of analog input channels of all HSL AI/O modules which are connected to the master card. When you want to stop the A/D conversion, please use this function to stop it.

_8154_db51_HSL_A_set_signal_range:

This function is used to set the input range of the specified HSL AI/O modules.

_8154_db51_HSL_A_get_signal_range:

This function is used to get the input range of the specified HSL AI/O modules.

_8154_db51_HSL_A_get_input_mode:

This function is used to get the signal input mode of HSL AI/O modules. This is determined by hardware jumper setting.

_8154_db51_HSL_A_set_last_channel:

This function is used to set the last number of analog input channels of HSL AI/O modules. For example, HSL-AI16AO2 has 16 analog input with single-ended wiring. If users just want to read back the first 4 analog input data of this module, assign the last channel as 3. The analog input channel index is from 0. As a result, the AI channel 0 to 4 would be enabled and the others would be disabled.

_8154_db51_HSL_A_get_last_channel:

This function is used to retrieve the last number of analog input channels of HSL AI/O modules. For example, if you use **_8154_db51_HSL_A_set_last_channel** and set the last channel as 5, then you can read the value with **_8154_db51_HSL_A_get_lastchannel**.

_8154_db51_HSL_A_read_input:

This function is used to read the specified AI channel of the slave module.

_8154_db51_HSL_A_write_output:

This function is used to write the specified AO channel of the slave module.

_8154_db51_HSL_A_read_output:

This function is used to read back the analog output data from the HSL AI/O modules with the specified the analog output channel.

_8154_db51_HSL_A_sync_rw:

This function is used to read AI data and write AO data at the specified channel synchronously of HSL AIO module. It can let users read/write the data at one time.

_8154_db51_HSL_A_get_version:

This function is to read the kernel version of the HSL AI/O modules.

@ Syntax

C/C++ (DOS, Windows 98/NT/2K/XP)

```
I16 _8154_db51_HSL_A_start_read (I16 card_ID);
I16 _8154_db51_HSL_A_stop_read (I16 card_ID);
I16 _8154_db51_HSL_A_set_signal_range (I16
     card_ID, I16 slave_No, I16 signal_range);
I16 _8154_db51_HSL_A_get_signal_range (I16
     card_ID, I16 slave_No, I16 *signal_range);
I16 _8154_db51_HSL_A_get_input_mode (I16 card_ID,
     I16 slave_No, I16 *mode);
I16 _8154_db51_HSL_A_set_last_channel (I16
     card_ID, I16 slave_No, I16 last_channel);
I16 _8154_db51_HSL_A_get_last_channel (I16
     card_ID, I16 slave_No, I16 *last_channel);
I16 _8154_db51_HSL_A_read_input (I16 card_ID, I16
     slave_No, I16 ai_channel, F64 *ai_data);
I16 _8154_db51_HSL_A_write_output (I16 card_ID,
     I16 slave_No, I16 ao_channel, F64 ao_data);
I16 _8154_db51_HSL_A_read_output (I16 card_ID,
     I16 slave_No, I16 ao_channel, F64 *ao_data);
I16 _8154_db51_HSL_A_sync_rw (I16 card_ID, I16
     slave_No, I16 ai_channel, F64 *ai_data, I16
     ao_channel, F64 ao_data);
I16 _8154_db51_HSL_A_get_version (I16 card_ID,
     I16 slave_No, I16 *ver);
```

Visual Basic (Windows 98/NT/2K/XP)

```
B_8154_db51_HSL_A_start_read (ByVal card_ID As
      Integer) As Integer
B_8154_db51_HSL_A_stop_read (ByVal card_ID As
      Integer) As Integer
B_8154_db51_HSL_A_set_signal_range (ByVal card_ID
      As Integer, ByVal slave_No As Integer, ByVal
      signal_range As Integer) As Integer
B_8154_db51_HSL_A_get_signal_range (ByVal card_ID
      As Integer, ByVal slave_No As Integer,
      signal_range As Integer) As Integer
B_8154_db51_HSL_A_get_input_mode (ByVal card_ID
      As Integer, ByVal slave_No As Integer, mode
      As Integer) As Integer
B_8154_db51_HSL_A_set_last_channel (ByVal card_ID
      As Integer, ByVal slave_No As Integer, ByVal
      last_channel As Integer) As Integer
B_8154_db51_HSL_A_get_last_channel (ByVal card_ID
      As Integer, ByVal slave_No As Integer,
      last_channel As Integer) As Integer
B_8154_db51_HSL_A_read_input (ByVal card_ID As
      Integer, ByVal slave_No As Integer, ByVal
      ai_channel As Integer, ai_data As Double) As
      Integer
B_8154_db51_HSL_A_write_output (ByVal card_ID As
      Integer, ByVal slave_No As Integer, ByVal
      ao_channel As Integer, ByVal ao_data As
      Double) As Integer
B_8154_db51_HSL_A_read_output (ByVal card_ID As
      Integer, ByVal slave_No As Integer, ByVal
      ao_channel As Integer, ao_data As Double) As
      Integer
B_8154_db51_HSL_A_sync_rw (ByVal card_ID As
      Integer, ByVal slave_No As Integer, ByVal
      ai_channel As Integer, ai_data As Double,
      ByVal ao_channel As Integer, ByVal ao_data
      As Double) As Integer
B_8154_db51_HSL_A_get_version (ByVal card_ID As
      Integer, ByVal slave_No As Integer, ver As
      Integer) As Integer
```

@ Argument

`card_ID`: Specify the HSL master card index. Normally, the board index sequence would be decided by the system. The index is from 0.

`slave_No`: Specifiy the slave module with slave index which want to perform this function. The valid value is from 1 to 63.

`signal_range`: The single range of analog input setting.

For HSL-AI16AO2-M-VV:

- 0: ± 1.25 V
- 1: ± 2.5 V
- 2: ± 5 V
- 3: ± 10 V

For HSL-AI16AO2-M-AV

- 0: ± 5mA
- 1: ± 10mA
- 2: ± 20mA
- 3: ± 20mA

`*signal_range`: Read back the single range of analog input setting.

For HSL-AI16AO2-M-VV

- 0: ± 1.25 V
- 1: ± 2.5 V
- 2: ± 5 V
- 3: ± 10 V

For HSL-AI16AO2-M-AV

- 0: ± 5mA
- 1: ± 10mA
- 2: ± 20mA
- 3: ± 20mA

***mode**: Set the mode

- 0: differential type
- 1: single-ended input.

last_channel: For single-ended setting, the maximum last channel is 15. For differential setting, the maximum last channel is 7.

***last_channel**: Obtain the last channel depending on what you set previously. For single-ended setting, the maximum last channel is 15. For differential setting, the maximum last channel is 7.

ai_channel: Specify the AI channel of the slave module which want to perform this function. The valid value is described as follows.

For HSL-AI16AO2-M-VV/AV:

- Differential: 0 - 15
- Single-ended: 0 - 7

ao_channel: Specify the AI channel of the slave module which want to perform this function. For HSL-AI16AO2-M-VV/AV, the valid value is 0 and 1.

***ai_data**: The AI data of the specified channel. The unit is Volt for HSL-AI16AO2-M-VV module and mA for HSL-AI16AO2-M-AV module.

ao_data: The AO data of the specified channel in unit of volts.

***ver**: kernel version number.

@ Return Code

```
ERR_No_Error
ERR_Invalid_Board_Number
ERR_Connect_Index
ERR_Time_Out
ERR_Memory_Mapping
ERR_Satellite_Number
ERR_Satellite_Type
ERR_Over_Max_Address
ERR_AI16AO2_Signal_Range
```

Appendix A Scan Time Table

A.1 Full Duplex Mode

Slave Index Number	Cycle Time under 3 Mbps	Cycle Time under 6 Mbps	Cycle Time under 12 Mbps
Base Unit	60.67 µs	30.33 µs	15.17 µs
< 3(*)	182.00µs	91.00 µs	45.50 µs
5	303.33 µs	151.67 µs	75.83 µs
10	606.67µs	303.33 µs	151.67 µs
20	1.213 ms	606.67 µs	303.33 µs
30	1.820 ms	910.00 µs	455.00 µs
40	2.427 ms	1.213 ms	606.67 µs
50	3.033 ms	1.516 ms	758.33 µs
60	3.640 ms	1.820 ms	910.00 µs
63	3.822 ms	1.911 ms	955.50 µs

(*) The minimum scan time for full duplex mode at different transmission speeds.

A.2 Half Duplex Mode

Slave Index Number	Cycle Time under 3 Mbps	Cycle Time under 6 Mbps	Cycle Time under 12 Mbps
Base Unit	118 µs	59 µs	29.5 µs
< 3(*)	354 µs	177 µs	88.5 µs
5	590 µs	295 µs	147.5 µs
10	1.180 ms	590 µs	295 µs
20	2.360 ms	1.180 ms	590 µs
30	3.540 ms	1.770 ms	885 µs
40	4.720 ms	2.360 ms	1.180 ms
50	5.900 ms	2.950 ms	1.475 ms
60	7.080 ms	3.540 ms	1.770 ms
63	7.434 ms	3.717 ms	1.859 ms

(*) The minimum scan time for half duplex mode at different transmission speeds.

Appendix B HSL-AI16AO2 Calibration

B.1 Before you proceed

Before calibrating the HSL-AI16AO2-M-VV and HSL-AI16AO2-M-AV, take note of the following:

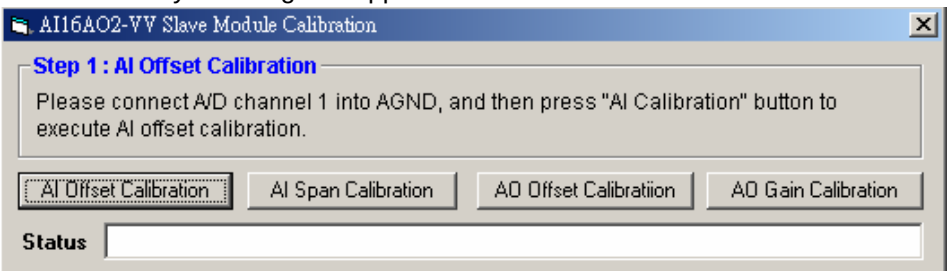
1. Make sure that the signal type is single-ended. You may set this via the jumper.
2. Use a precise calibrator that can generate a precise 5 V.
3. Check the status text to know if the calibration is successful or not.
4. Refer to the analog input field configuration below.

Single-ended mode																
Terminal No.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Signal Name	AI0	AI1	AI2	AI3	AI4	AI5	AI6	AI7	AI8	AI9	AI10	AI11	AI12	AI13	AI14	AI15
Terminal No.	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Signal Name	AO0	AO1	AGND	AGND	AGND	AGND	AGND	AGND	AGND	AGND	AGND	AGND	AGND	AGND	AGND	AGND

B.2 Calibrating the modules

To calibrate the modules:

1. Press the Calibration button from the HSL-AI16AO2 utility. A dialog box appears.



2. Connect AI channel 1 to AGND. The AI channel index is from 0 to 15. Take note of the index. After wiring, press the AI Offset Calibration button.
3. Connect AI channel 0 to the calibrator, then, press the AI Span Calibration button.
4. Connect AI channel 12 to AO channel 0, and AI channel 14 to AO channel 1, then press the AO Offset Calibration button.
5. If the previous step is successful, press the AO Gain Calibration button to finish the calibration.

If calibration is successful, the module is ready for use. If not, check the wiring and calibrator, then repeat the calibration procedures.

Appendix C HSL-HUB/Repeater Information

C.1 Recommended transfer rates, total extension distance, and number of installed HSL-HUB/Repeater

Transmission rate	Number of inserted Hubs (Repeater)							
	Basic configuration	1	2	3	4	5	6	7
3Mbps	300 m	600 m	900 m	1.2 km	1.5 km	1.8 km	2.1 km	2.4 km
6Mbps	200 m	400 m	600 m	800 m	1 km	1.2 km	1.4 km	1.6 km
12Mbps	100 m	200 m	300 m	400 m	500 m	600 m	700 m	800 m

C.2 Scan time table

C.2.1 Full duplex/12 Mbps

Number of inserted Hubs (Repeater)	Slave Index Number		
	3 (Min.)	30	63 (max)
Basic configuration (0)	45.50 us	455.00 us	955.50 us
1	82.00 us	820.00 us	1722.00 us
2	118.00 us	1180.00 us	2478.00 us
3	154.00 us	1540.00 us	3234.00 us
4	190.00 us	1900.00 us	3990.00 us
5	226.00 us	2260.00 us	4746.00 us
6	262.00 us	2620.00 us	5502.00 us
7	298.00 us	2980.00 us	6258.00 us

C.2.2 Full duplex/6 Mbps

Number of inserted Hubs (Repeater)	Slave Index Number		
	3 (Min.)	30	63 (max)
Basic configuration (0)	91.00 us	910.00 us	1911.00 us
1	164.00 us	1640.00 us	3444.00 us
2	236.00 us	2360.00 us	4956.00 us
3	308.00 us	3080.00 us	6468.00 us
4	380.00 us	3800.00 us	7980.00 us
5	452.00 us	4520.00 us	9492.00 us
6	524.00 us	5240.00 us	11004.00 us
7	596.00 us	5960.00 us	12516.00 us

C.2.3 Full duplex/3 Mbps

Number of inserted Hubs (Repeater)	Slave Index Number		
	3 (Min.)	30	63 (max)
Basic configuration (0)	182.00 us	1820.00 us	3822.00 us
1	328.00 us	3280.00 us	6888.00 us
2	472.00 us	4720.00 us	9912.00 us
3	616.00 us	6160.00 us	12936.00 us
4	760.00 us	7600.00 us	15960.00 us
5	904.00 us	9040.00 us	18984.00 us
6	1048.00 us	10480.00 us	22008.00 us
7	1192.00 us	11920.00 us	25032.00 us