

---

# *The I-7530 RS-232/CAN Converter*

## User's Manual

### Warranty

All products manufactured by ICP DAS are under warranty regarding defective materials for a period of one year from the date of delivery to the original purchaser.

### Warning

ICP DAS assumes no liability for damages resulting from the use of this product. ICP DAS reserves the right to change this manual at any time without notice. The information furnished by ICP DAS is believed to be accurate and reliable. However, no responsibility is assumed by ICP DAS for its use, or for any infringements of patents or other rights of third parties resulting from its use.

### Copyright

Copyright 1997 by ICP DAS. All rights are reserved.

### Trademark

The names used for identification only may be registered trademarks of their respective companies.

---

## Table of Contents

<b>1. Introduction</b>	<b>3</b>
1.1 Features	4
1.2 Specifications	4
<b>2. Hardware</b>	<b>5</b>
2.1 Block Diagram	5
2.2 Pin Assignment	6
2.3 Hardware connection	7
2.4 Terminator resistor settings	8
2.5 Init/Normal Dip-switch	9
2.6 LED Indication	10
2.7 Cable selection	11
<b>3. Software Utility</b>	<b>12</b>
3.1 How to configure the module parameters	13
3.2 How to test the module transmission performance	15
<b>4. Command list</b>	<b>17</b>
4.1 tIIILDD...[CHK]<CR>	18
4.2 TIIIL[CHK]<CR>	18
4.3 eIIIIIIILDD...[CHK]<CR>	19
4.4 EIIIIIIIL[CHK]<CR>	19
4.5 S[CHK]<CR>	20
4.6 C[CHK]<CR>	22
4.7 P0BBDSPA[E][CHK]<CR>	22
4.8 P1B [CHK]<CR>	23
4.9 RA[CHK]<CR>	24
4.10 General Error code for all command	25
<b>5. Troubleshooting</b>	<b>25</b>

---

# 1. Introduction

CAN (Controller Area Network) is a serial bus control protocol especially suited to structure intelligent industry devices networks and build smart automatic control systems. Figure 1 shows the application architecture for I-7530 modules. The PC can be the CAN host, monitor or HMI to access/control the CAN device through the CAN network by the I-7530 Converter. The programmable RS-232 device (For example: I-8411/I-8431/I-8811/I-8831/W-8031/W-8331/W-8731 embedded controller) can use the serial port to connect to the CAN network via the I-7530 module.

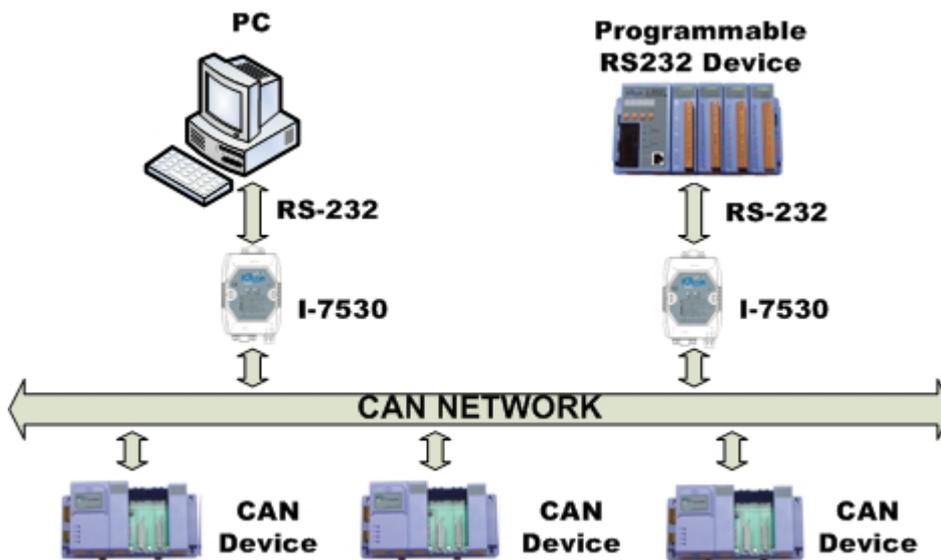


Figure 1: Application of I-7530

In order to use the CAN network with traditional RS-232 devices, we provide a way to achieve this goal. Therefore, I-7530 is designed to unleash the power of the CAN bus via the RS-232 communication method. It accurately converts messages between CAN and RS-232 networks. It also allows users to communicate with CAN devices easily through any PC or devices with a RS-232 interface. The main features and specification of this are described as below:

---

## 1.1 Features

- Microprocessor inside with 20MHz;
- CAN interface connector: D-Sub 9-pin;
- Phillip 82C250 CAN transceiver;
- Provide One CAN and RS-232 Port;
- Built in CAN/RS-232 Converter firmware;
- Max transmission speed up to 1M bps for CAN and 115.2K bps for RS-232 interface;
- Provide Software configuration for CAN and RS-232 communication;
- Max transmission distance over 1000m based on CAN specification, sec section 2.7;
- Support both CAN 2.0A and CAN 2.0B;
- Build-in jumper to provide 120 ohm terminal resistor or not in CAN network;
- Provide power, data flow and error LED indicator for CAN and RS-232;
- Built-in a Dual-Watchdog function in the module;

## 1.2 Specifications

### RS-232 specs:

- RS-232 interface connector: D-sub 9-pin female;
- RS-232 Baud Rate: 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 and 230400 (not guaranteed);

### CAN specs:

- CAN interface connector: D-sub 9-pin male;
- CAN Baud Rate: 10K, 20K, 50K, 100K, 125K, 250K, 500K, 800K and 1M bps;
- Isolation voltage: 3000Vrms on the CAN side;

### Power requirement:

- Unregulated +10V DC ~ +30V DC;
- Power reverse protection, Over-Voltage brown-out protection;
- Power consumption: 1W;

### Module specs:

- Dimensions: 123mm x 72mm x 33mm;
- Operating temperature: -25 to 75°C (14 to 185°F);

- 
- Storage temperature: -25 to 80°C (-13 to 185°F);
  - Humidity: 5 to 95%, non-condensing;
  - LEDs: ON LED for power and data flow;  
ERR LED for error;

Software Utility tool:

- CAN bus Baud Rate configuration;
- CAN acceptance filter configuration;
- Setting for CAN 2.0A or 2.0B working mode;
- Setting the Baud Rate and data format on RS-232;
- Setting the RS-232 communication with checksum function or not;
- Provides a response message or not when the ERR LED is turned on;
- Provides a quick testing function for transmitting/receiving CAN messages;

Application:

- Factory Automation;
- Building Automation;
- Home Automation;
- Control system;
- Monitor system;
- Vehicle Automation;

## 2. Hardware

### 2.1 Block Diagram

Figure 2 is a block diagram illustrating the functions on the I-7530 module. It provides the 3000Vrms Isolation in the CAN interface site. And hardware media in RS-232 interface is only adopted 3-wire connection.

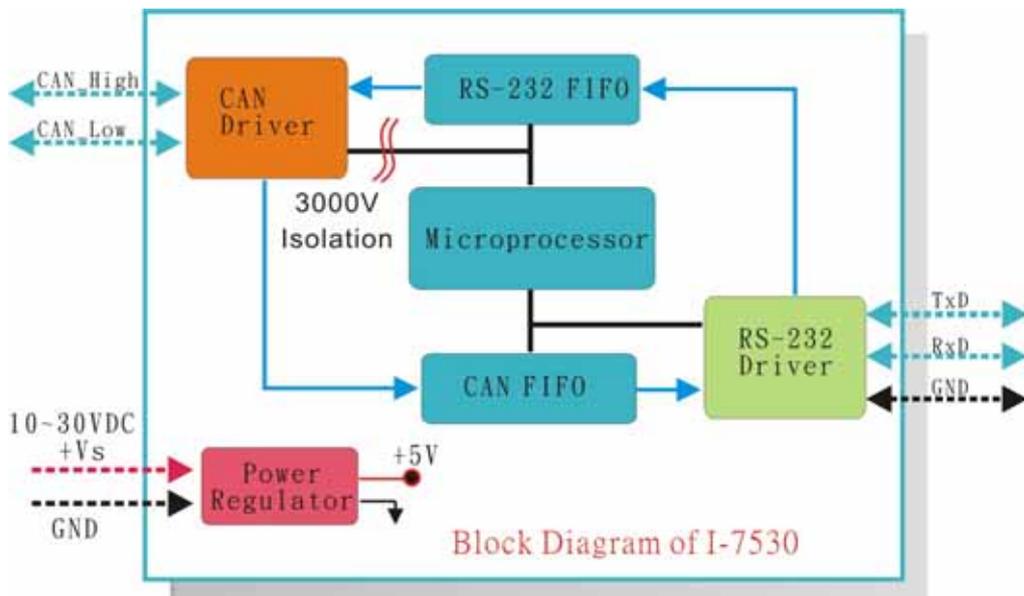


Figure 2: Block diagram of I-7530

## 2.2 Pin Assignment

Table 1: RS-232 DB9 Female Connector (CN1)

Terminal	3-wire RS-232
1	Not Connect
2	<b>TXD</b>
3	<b>RXD</b>
4	Not Connect
5	<b>GND</b>
6	
7	
8	
9	

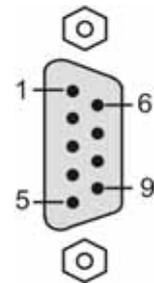
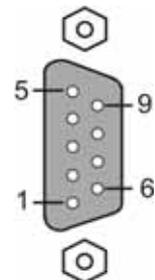


Table 2: CAN DB9 Male Connector (CN2)

Terminal	2-wire CAN
1	Not Connect
2	<b>CAN Low</b>
3	
4	
5	Not Connect
6	
7	<b>CAN High</b>
8	
9	Not Connect



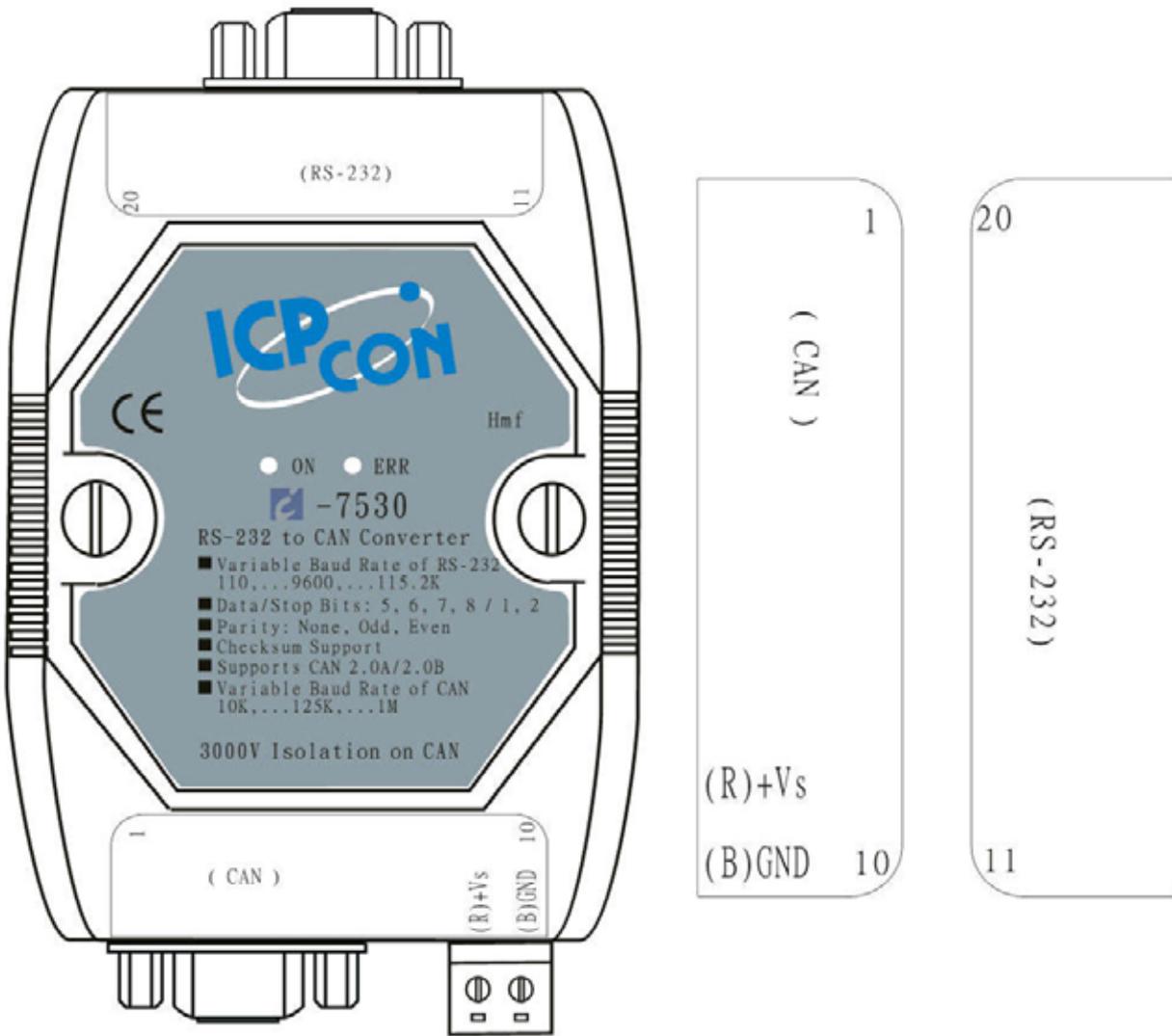


Figure 3: Pin Assignment on the I-7530

### 2.3 Hardware connection

The RS-232 port on the I-7530 (DB9 female) is inserted directly into a PC's COM serial port or via a cable to the Host system.



Figure 4: RS-232 connection

The pin assignment of the CAN port on the I-7530 (DB9 male) defined in both the CANopen DS102 profile and in appendix C of the DeviceNet specifications. It is the standard pin assignment for CAN.

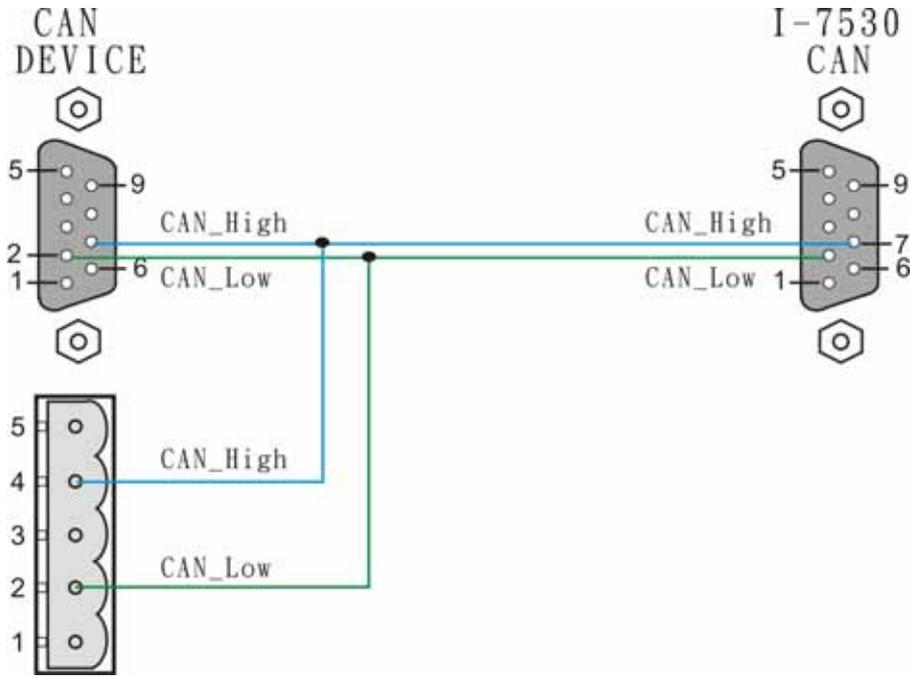


Figure 5: CAN connection

## 2.4 Terminator resistor settings

According to the ISO 11898 specifications, the CAN Bus network must be terminated by two termination resistances ( $120\ \Omega$ ) for proper operation, as shown in the below figure.

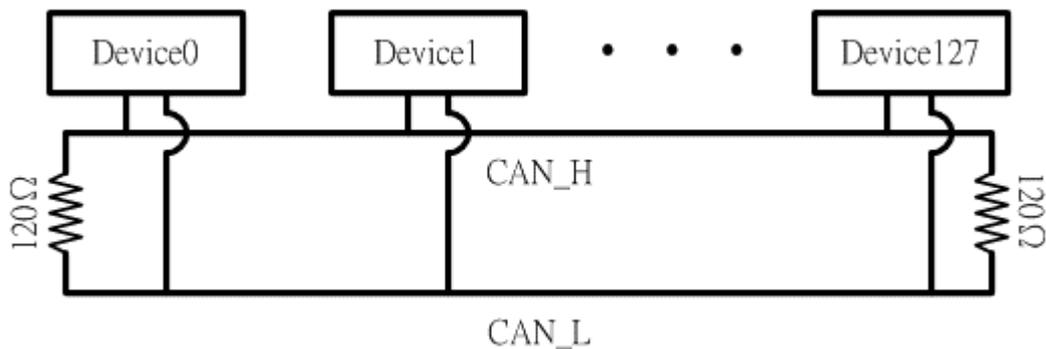


Figure 6: Terminator resistor

---

Therefore, the I-7530 module supplies a jumper for users to connect the terminator resistor or not. If users want to use this terminator resistor, please open the I-7530 cover and use the JP3 jumper to activate the 120 terminator resistor built in the system, as in the following figure. Note that the default setting is active.

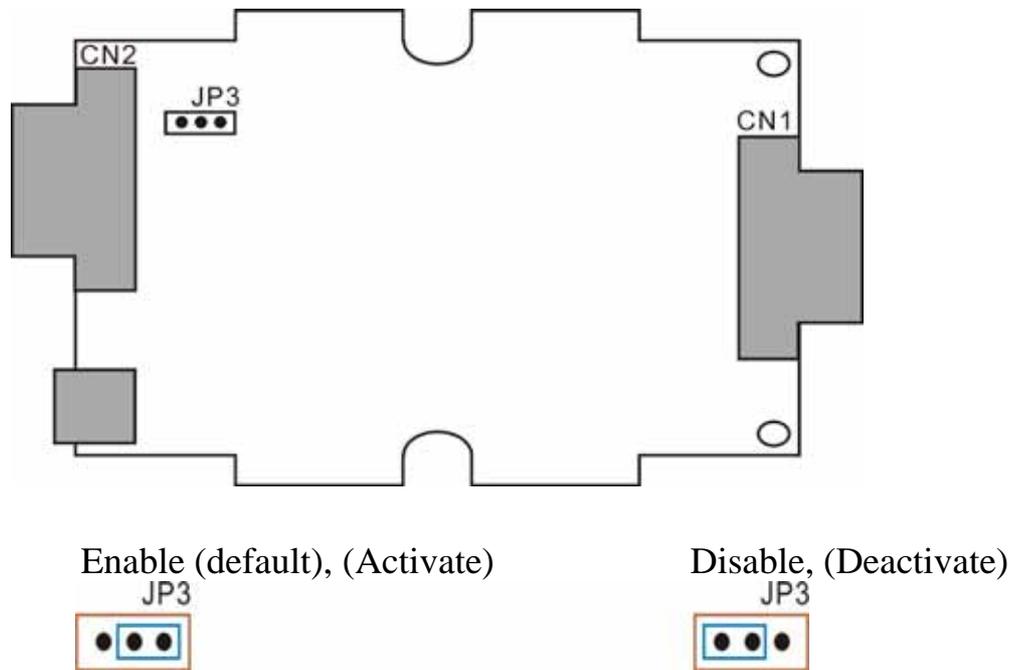


Figure 7: JP3 positions

## 2.5 Init/Normal Dip-switch

On the back of the I-7530 module, there is a dip-switch used for setting the operation or configuration mode of the module function. In the normal situation, the user needs to first make a configuration in order to provide the correct function when the module works in the operation mode. The following steps show how to use this dip-switch.

- (1) Configuration mode: Please first set the dip-switch to the “Init” (Initial) position. Then the I-7530 will work in the configuration mode after the power for the module has been turned on. In this case, users can configure the communication parameters of the I-7530 module by using the I-7530 Utility tool.
- (2) Operation mode: After the configuration, users need to set the dip-

---

switch to the “Normal” position. Users need to turn the power off then on again so that they can use the I-7530 in the operation mode. Once you have completed your configurations and have switched to the operation mode, then messages can pass between the CAN and the RS-232. This transmission process depends on the configuration parameters that the users have configured. That is, the I-7530 functions as a RS-232/CAN converter.

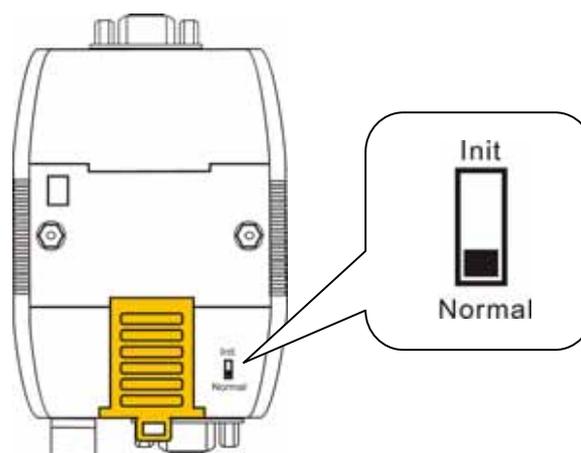


Figure 8: Dip-switch of I-7530

## 2.6 LED Indication

The I-7530 provides the Converter function between the RS-232 port and the CAN port. It can handle both an 11-bit and 29-bit ID format according to whether it is a CAN 2.0A or 2.0B. In order to provide high performance when converting data, the I-7530 has built in software FIFO queues, which include 1000 CAN data frames. It also provides two LEDs to indicate to users what situation the I-7530 is in. The ON LED indicator establishes the power and operation situation simultaneously. If the module is working in the operation mode, the ON LED is always turned on. However, when the module is working in the configuration mode, the ON LED will flash approximately once per second. The ERR LED is used for demonstrating an error that has occurred. That is, the ERR LED is normally turned off when the module works in a good condition. On the other hand, when the BUS OFF error indicator of the CAN controller has been alerted, the ERR LED will turn on. In this case, users can only reboot the I-7530 module in order to turn off the ERR LED. If the CAN/RS232 FIFO overflows, the ERR LED will also be turned on.

Furthermore, If users want to check what the error situation is they must send the command string “S[CHK]<CR>”. Depending on the result, users will need to either reboot the I-7530 or use the command string “C[CHK]<CR>” to clear the FIFO flag, which will then turn off the ERR LED.



Figure 9: LED position

Table 3: LED indication

LED Name	I-7530 Condition	LED Status
ON LED	Configuration Mode	Flashes once per second
	Operation Mode	Always turned on
ERR LED	Some errors occurred	Always turned on

## 2.7 Cable selection

The CAN bus is a Balanced (differential) 2-wire interface running over either a Shielded Twisted Pair (STP), Un-shielded Twisted Pair (UTP), or Ribbon cable. The CAN-L and CAN-H Wire start on one end of the total CAN network that a terminator of 120 Ohm is connected between CAN-L and CAN-H. The cable is connected from CAN node to CAN node, normally without or with short T connections. On the other end of the cable again a 120 (Ohm) terminator resistor is connected between the CAN lines. How to decide a cable type, cable length, and terminator depends on the baud rate in the CAN BUS network, please refer to the following table 4.



Figure 10: Un-shielded Twisted Pair (UTP)

Table 4: Cable selection

Bus speed	Cable type	Cable resistance/m	Terminator	Bus Length
50k bit/s at 1000m	0.75~0.8mm <sup>2</sup> 18AWG	70 mOhm	150~300 Ohm	600~1000m
100k bit/s at 500m	0.5~0.6 mm <sup>2</sup> 20AWG	< 60 mOhm	150~300 Ohm	300~600m
500k bit/s at 100m	0.34~0.6mm <sup>2</sup> 22AWG, 20AWG	< 40 mOhm	127 Ohm	40~300m
1000k bit/s at 40m	0.25~0.34mm <sup>2</sup> 23AWG, 22AWG	< 40 mOhm	124 Ohm	0~40m

Note: The AWG means a standard method used to measure wire. The numbering system works backwards from what people would think, the thicker (heavier) the wire, the lower the number. For example: a 24AWG wire is thicker/heavier than a 26AWG wire.

### 3. Software Utility

Users can download I-7530 Utility software from the ICP DAS web site (<http://www.icpdas.com/products/7000/i-7530.htm>) or get this information from the ICP DAS companion CD-ROM. The I-7530 Utility tool can be used to configure the operation condition between the CAN and RS-232 communications. Also it can be used to transmit or receive a CAN message for simple testing of the module's functions. To start the "I-7530 Utility", please run the I7530.exe file. The screenshot of the startup screen for this Utility is given in the below figure. Connect the I-7530's COM port with the PC's COM port via a RS-232 cable. Then the user can connect the CAN interface into the CAN network based on the CAN specifications. For further information related to this, please refer to section 1.5 of this manual on how to make a hardware connection.

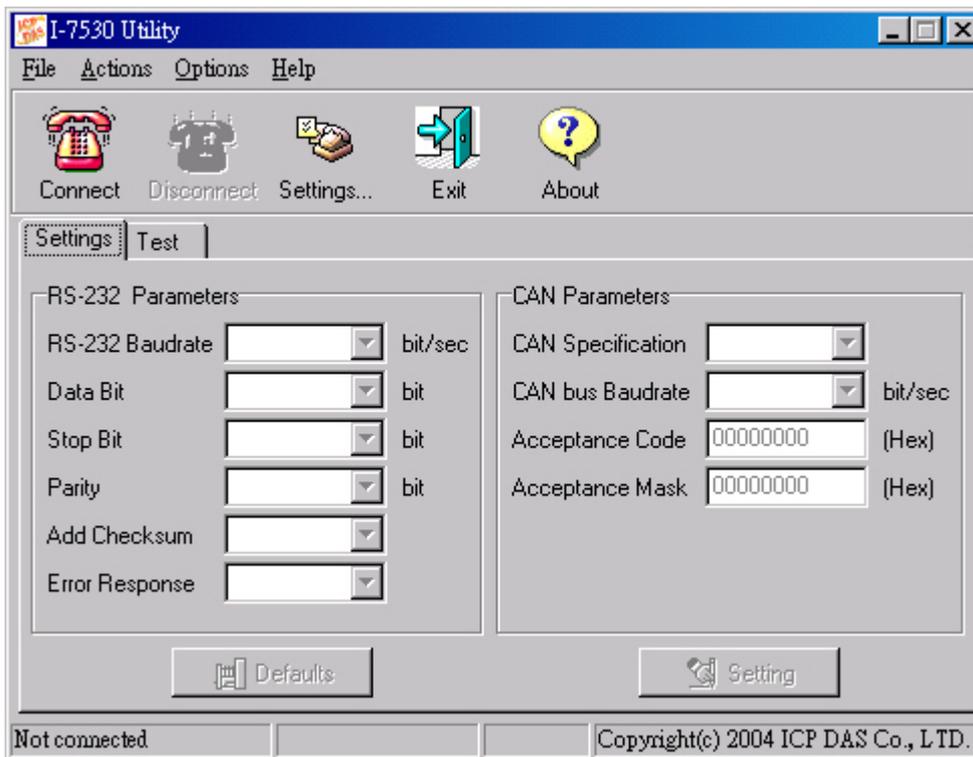


Figure 11: I-7530 Utility

### 3.1 How to configure the module parameters

The following procedure will guide you on how to configure the communication parameters for the CAN and RS-232 interface. The configuration steps are depicted as below:

1. Set the Init/Normal switch to the Init position, which is located at the back of the I-7530 module.
2. Connect the power source (the 10~30 DC volts) into the I-7530 module.
3. The I-7530 module's ON LED will flash approximately once per second. That means the I-7530 module is working in the configuration mode.
4. The user must run the I-7530's Utility software after they have made a wire connection between the PC and the I-7530 via the COM port.
5. Click the "Settings..." icon on the I-7530 Utility tool bar in order to select which PC COM port will be used to connect to the I-7530 COM port, as shown in the following figure. Then click the "OK" button.

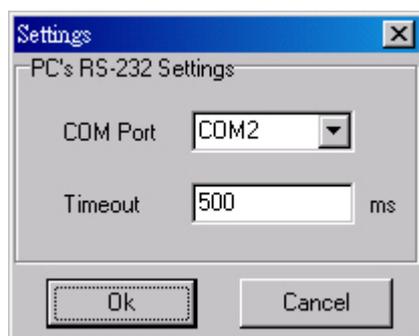


Figure 12: COM port setting

6. Click the “Connect” icon to bring out the I-7530 configuration window. The I-7530 Utility will show the communication information from the I-7530 module in the window, as shown in the following figure.

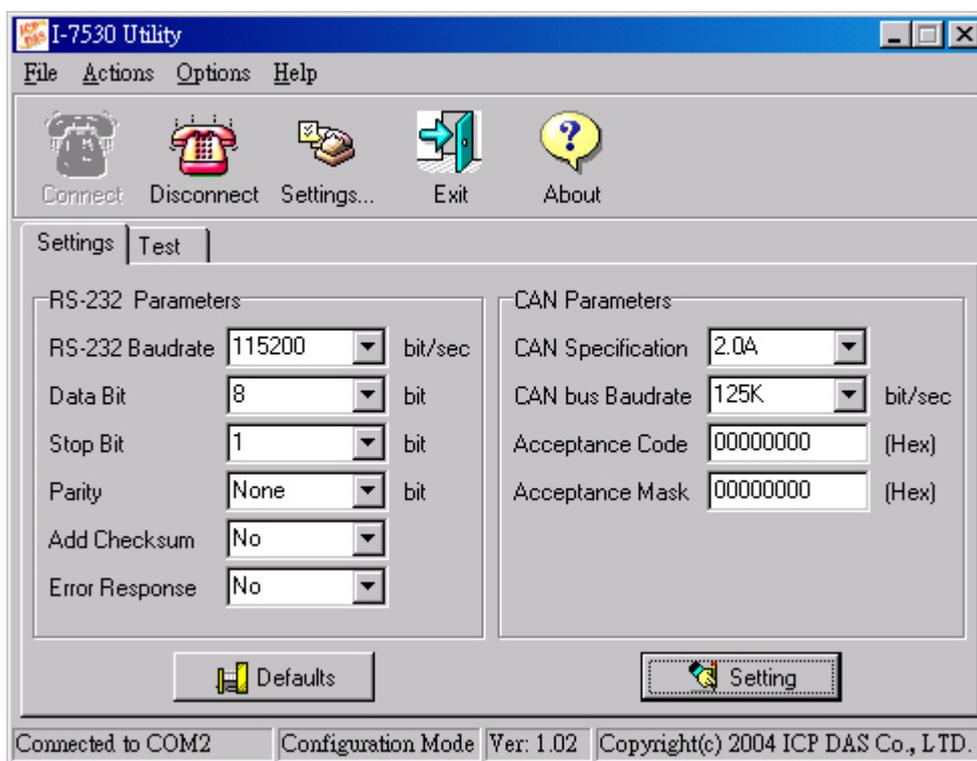


Figure 13: Connect to the configuration mode of the I-7530

7. Choose the “Settings” tab to open the configuration window for the CAN and the RS-232 communication parameters of the I-7530. Once users have finished changing the parameter settings, please click the “Setting” button to store the communication parameters into the EEPROM on the I-7530.

Note: If users click the “Defaults” icon, all of the CAN and RS-232 communication parameters on the I-7530 will be set to the default

---

values, which are:

RS232:    RS-232 Baudrate    = 115200  
          Data Bit                = 8  
          Stop Bit                = 1  
          Parity                    = None  
          Add Checksum        = No  
          Error Response       = No

CAN:        CAN Specification    = 2.0A  
          CAN bus Baudrate    = 125K  
          Acceptance Code    = 00000000  
          Acceptance Mask    = 00000000

### 3.2 How to test the module transmission performance

The following procedure will guide you to learning how to transmit/receive CAN messages to/from other devices/PCs by using the I-7530 converter. Note that users can only employ this test function for the system if they are using the default communication parameters for the RS-232 as provided above.

1. Set the Init/Normal switch to the Normal position, which is found at the back of the I-7530 module.
2. Connect the I-7530's CAN port into the CAN network, which must at least have one CAN device on the network.
3. Supply the 10~30 volts DC source into the I-7530 module through the power terminal.
4. The ON LED on the I-7530 module will be turned on. That means the I-7530 is working in the operation mode.
5. Run the I-7530 Utility software after there is a wire connection as shown in Section 1.5.
6. Click the "Settings..." icon on the tool bar in the I-7530 Utility to configure the PC COM port, which will be used to connect to the COM port of the I-7530.

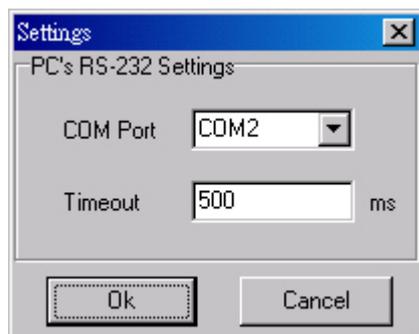


Figure 14: COM port setting

- Click the “Connect” icon to connect to the I-7530’s testing mode. Then the I-7530 Utility will show the test tab in the window as can be seen in the below figure.

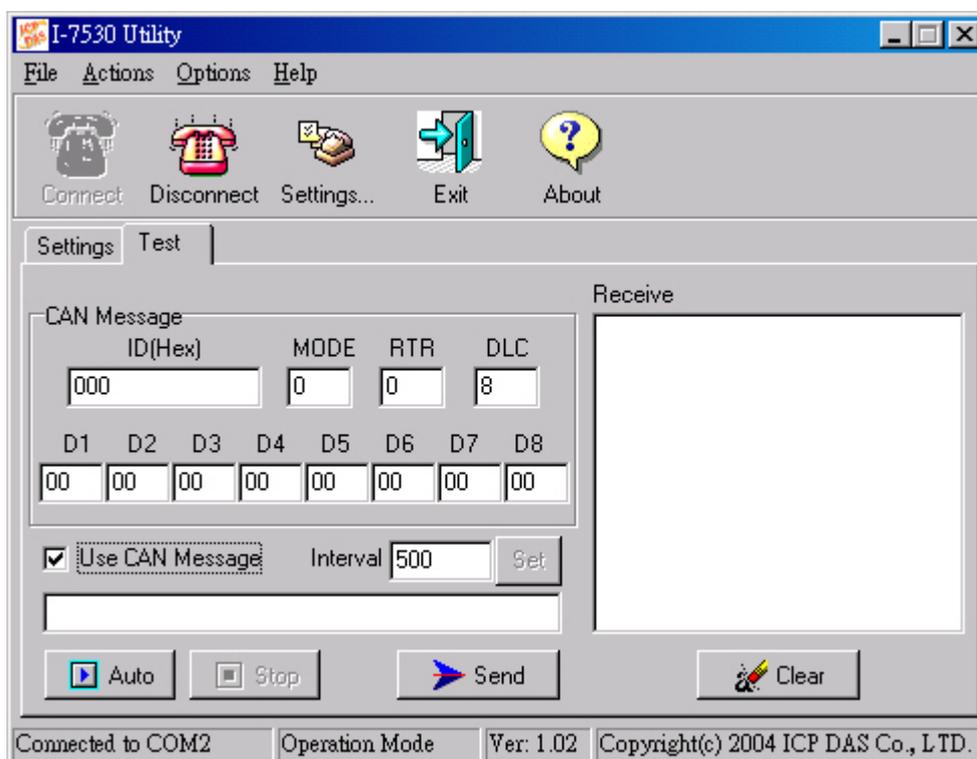


Figure 15: Connect to the operation mode of I-7530

- Select the “Test” tab. Users will then be able to transmit or receive CAN messages via the I-7530 module. There are two methods for communicating with their CAN. The first method (check “Use CAN Message”) requires users to understand their CAN specifications. And input the correct data in the CAN message format. The second method (uncheck “Use CAN Message”) allows the use of the command string found in table 5 to transmit messages. Both methods require the user to click the “Send” button to transmit the information into the CAN network.

The CAN message will then automatically be displayed in the “Receive” list box. Besides, user can press the “Clear” button to remove the messages on the list box and then the new CAN messages will be shown on the list box again.

## 4. Command list

For easy application, we provide 9 command strings to allow users to send and receive commands and responses through the I-7530. It can cover most applications of different requests. The general format of the commands for the I-7530 are given below:

Command Format: <Command>[CHK]<CR>

- <Command> : The RS-232 commands of the I-7530 Converter.
- [CHK] : 2-character checksum value. It is effective only if the checksum mechanism is set to enable by using I-7530 Utility.
- <CR> : All RS-232 commands of the I-7530 Converter must end with the character “<CR>” (The ASCII value is 13).

The 9 command formats are given in the following table. More detailed information related to of the each command will be described in the following sub sections.

Table 5: Command list table

Command	Description
tIIILDD...[CHK]<CR>	Send or receive a standard data frame.
TIIIL[CHK]<CR>	Send or receive a standard remote frame.
eIIIIIIILDD...[CHK]<CR>	Send or receive an extended data frame.
EIIIIIIIL[CHK]<CR>	Send or receive an extended remote frame.
S[CHK]<CR>	Read the status value of I-7530
C[CHK]<CR>	Clear CAN/RS232 FIFO error flag of overflow.
P0BBDSPA[E][CHK]<CR>	Change the RS-232 configuration
P1B [CHK]<CR>	Change the CAN configuration
RA[CHK]<CR>	Reboot the I-7530 module.

---

#### 4.1 tIIILDD...[CHK]<CR>

**Description:** Send or receive a standard CAN data frame.

➤ **Syntax:** tIIILDD...[CHK]<CR>

<b>t</b>	Represent a standard (2.0A) data frame.
<b>III</b>	11 bits Identifier (000~7FF)
<b>L</b>	Data length (0~8)
<b>DD...</b>	Input data frame value according to the data length (00~FF)

➤ **Response:** Valid command: No response

Invalid command: ?<Error Code><CR>

➤ **Note:** It is necessary to enable the “Error Response” function in the I-7530 Utility, in order to receive Syntax and/or communication error information at the host PC.

➤ **Example:**

Command: t03F6112233445566<CR>

Send a CAN message with a standard data frame. ID=03F, DLC=6, data1=11, data2=22, data3=33, data4=44, data5=55 and data6=66.

#### 4.2 TIIIL[CHK]<CR>

**Description:** Send or receive a standard CAN remote frame.

➤ **Syntax:** TIIIL[CHK]<CR>

<b>T</b>	Represents a standard (2.0A) remote frame.
<b>III</b>	11 bits Identifier (000~7FF)
<b>L</b>	Data length (0~8)

➤ **Response:** Valid command: No response

Invalid command: ?<Error Code><CR>

➤ **Note:** It is necessary to enable the “Error Response” function in the I-7530 Utility, in order to receive Syntax and/or communication error information at the host PC.

---

➤ **Example:**

Command: T2E88<CR>

Send a CAN message with a standard remote frame. ID=2E8,  
DLC=8.

#### 4.3 eIIIIIIILDD...[CHK]<CR>

**Description:** Send or receive an extended CAN data frame.

➤ **Syntax: eIIIIIIILDD...[CHK]<CR>**

<b>e</b>	Stands for the extended (2.0B) data frame.
<b>IIIIIIII</b>	29 bits Identifier (00000000~1FFFFFFF)
<b>L</b>	Data length (0~8)
<b>DD...</b>	Input data frame value according to the data length (00~FF)

➤ **Response:** Valid command: No response

Invalid command: ?<Error Code><CR>

➤ **Note:** It is necessary to enable the “Error Response” function in the I-7530 Utility, in order to receive Syntax and/or communication error information at the host PC.

➤ **Example:**

Command: e1234567851122334455<CR>

Send a CAN message with an extended data frame. ID=12345678,  
DLC=5, data1=11, data2=22, data3=33, data4=44 and data5=55.

#### 4.4 EIIIIIIIL[CHK]<CR>

**Description:** Send or receive an extended CAN remote frame.

➤ **Syntax: EIIIIIIIL[CHK]<CR>**

<b>E</b>	Stands for the extended (2.0B) CAN remote frame.
<b>IIIIIIII</b>	29 bits Identifier (00000000~1FFFFFFF)
<b>L</b>	Data length (0~8)

- **Response:** Valid command: No response  
Invalid command: ?<Error Code><CR>
- **Note:** It is necessary to enable the “Error Response” function in the I-7530 Utility, in order to receive Syntax and/or communication error information at the host PC.
- **Example:**  
Command: E010156786<CR>  
Send a CAN message with an extended remote frame. ID=01015678, DLC=6.

#### 4.5 S[CHK]<CR>

**Description:** Read the I-7530 CAN Baud Rate and error flag message.

- **Syntax:** S[CHK]<CR>

**S** Command character.

- **Response:** Valid Command: !CFFTTRRO[CHK]<CR>  
Invalid command: ?<Error Code>[CHK]<CR>

**!** Delimiter for valid command  
**C** current baud rate setting of CAN  
**FF** CAN register  
**TT** CAN transmit error counter  
**RR** CAN receive error counter  
**O** CAN/RS-232 FIFO Overflow flag

- **Note:** It is necessary to enable the “Error Response” function in the I-7530 Utility, in order to receive Syntax and/or communication error information at the host PC. Furthermore, all response results are shown in the ASCII format. Users need to make an ASCII to hex format transformation in order to understand what the meaning is based on the following table.

Table 6: CAN baud rate list

AsciiToHex(C)	Description
0	10K baud rate of CAN
1	20K baud rate of CAN

2	50K baud rate of CAN
3	100K baud rate of CAN
4	125K baud rate of CAN
5	250K baud rate of CAN
6	500K baud rate of CAN
7	800K baud rate of CAN
8	1000K baud rate of CAN

Table 7: CAN register list

AsciiToHex(FF)	Description
Bit 7	Bus Off Mode
Bit 6	Error Passive Mode
Bit 5	Reserved
Bit 4	Overrun Buffer
Bit 3	Stuff Error General
Bit 2	CRC Error General
Bit 1	Form Error General
Bit 0	Acknowledgment Error General

Table 8: CAN/RS-232 FIFO overflow flag list

AsciiToHex(O)	Description
Bit 3	Reserved
Bit 2	Reserved
Bit 1	RS-232 FIFO Overflow
Bit 0	CAN FIFO Overflow

➤ **Example:**

Command: S<CR>

Receive: !50000000<CR>

Obtain some current information on the I-7530 module. The response will show the following results: CAN baud rate=250K, CAN register= normal, CAN transmit error counter=0, CAN receive error counter=0 and CAN/RS232 FIFO= normal.

---

#### 4.6 C[CHK]<CR>

**Description:** Clear the CAN/RS-232 FIFO overflows error flag on the module.

➤ **Syntax: C[CHK]<CR>**

**C** Command character.

- **Response:** Valid Command: No response. This will turn the ERR LED off after the ERR LED was turned on as a result of the CAN/RS232 FIFO overflow error flag.

Invalid command: ?<Error Code>[CHK]<CR>

- **Note:** It is necessary to enable the “Error Response” function in the I-7530 Utility, in order to receive Syntax and/or communication error information at the host PC.

➤ **Example:**

Command: C<CR>

#### 4.7 P0BBDSPAE[CHK]<CR>

**Description:** Change the RS-232 configuration on the I-7530 module and then reboot the I-7530 module.

➤ **Syntax: P0BBDSPAE[CHK]<CR>**

<b>P0</b>	Command character
<b>BB</b>	RS-232 Baud rate
<b>D</b>	Data bit
	0 = 5 bits Data formation
	1 = 6 bits Data formation
	2 = 7 bits Data formation
	3 = 8 bits Data formation
<b>S</b>	Stop bit (0=1 stop bit, 1=2 stop bits)
<b>P</b>	Parity (0=None, 1=Odd, 2=Even)
<b>C</b>	Checksum (0=No, 1=Yes)
<b>E</b>	Error Response (0=No, 1=Yes);

Table 9: RS-232 baud rate list

BB	Description
00	110 bps baud rate of RS-232
01	150 bps baud rate of RS-232
02	300 bps baud rate of RS-232
03	600 bps baud rate of RS-232
04	1200 bps baud rate of RS-232
05	2400 bps baud rate of RS-232
06	4800 bps baud rate of RS-232
07	9600 bps baud rate of RS-232
08	19200 bps baud rate of RS-232
09	38400 bps baud rate of RS-232
0A	57600 bps baud rate of RS-232
0B	115200 bps baud rate of RS-232
0C	230400 bps baud rate of RS-232

- **Response:** A valid command will write the RS-232 configuration parameters into the EEPROM and then reboot the I-7530 module.  
Invalid command: ?<Error Code><CR>
- **Note:** It is necessary to enable the “Error Response” function in the I-7530 Utility, in order to receive Syntax and/or communication error information at the host PC.
- **Example:**  
Command: P00B30000<CR>  
Set the RS232 baud rate=115.2K, data bit=8, stop bit=1, none parity, no checksum and No error responses into the I-7530 module and then reboot the I-7530 module.

#### 4.8 P1B [CHK]<CR>

**Description:** Change the CAN configuration of I-7530 module and then reboot the I-7530 module.

- **Syntax: P1B[CHK]<CR>**

**P1**            Command character  
**B**             CAN Baud rate

Table 10: CAN baud rate list

B	Description
0	10K baud rate of CAN
1	20K baud rate of CAN
2	50K baud rate of CAN
3	100K baud rate of CAN
4	125K baud rate of CAN
5	250K baud rate of CAN
6	500K baud rate of CAN
7	800K baud rate of CAN
8	1000K baud rate of CAN

- **Response:** A valid command will write the CAN configuration baud rate into the EEPROM and then reboot the I-7530 module.  
Invalid command: ?<Error Code><CR>
- **Note:** It is necessary to enable the “Error Response” function in the I-7530 Utility, in order to receive Syntax and/or communication error information at the host PC.
- **Example:**  
Command: P14<CR>  
Set the CAN baud rate=125K into the I-7530 module and then reboot the I-7530 module.

#### 4.9 RA[CHK]<CR>

**Description:** Reboot the I-7530 module. If the module is displaying on but the CAN bus is off, users can use this command to reboot the module in order to allow it to work in order again.

- **Syntax: RA[CHK]<CR>**

**RA**            Command character

- **Response:** Valid command will reboot the I-7530 module.  
Invalid command: ?<Error Code><CR>
- **Note:** It is necessary to enable the “Error Response” function in the I-7530 Utility, in order to receive Syntax and/or communication error information at the host PC.

➤ **Example:**

Command: RA<CR>

The I-7530 module will reboot after it had received this command.

#### 4.10 General Error code for all command

Table 11: Error code table

AsciiToHex(Error code)	Description
1	The head character of the command string is invalid.
2	The length of the command string is invalid.
3	The checksum of the command string is invalid.
4	Reserved
5	The RS-232 receiver is timeout.

## 5. Troubleshooting

If the Error response function on the I-7530 module is set to be “Yes”(that means enable) via the I-7530 Utility during the configuration period, the I-7530 will send the error code to the RS-232 device of the host PC through the RS-232 media when the I-7530 produces an error message automatically during the operation mode. The meanings of these error codes are given below:

Table 12: Error code table

Error code	Description	Possible causes & solutions
1	Invalid header	The RS-232 command string header is not “t”, “T”, “e”, “E”, “S”, “C”, “P0”, “P1” nor “RA”.
2	Invalid length	The data byte of the CAN Message does not match the data length of the CAN Message. For example: Error: t001512345<CR> Right: t00150102030405<CR>

---

3	Invalid checksum	The checksum from the RS-232 command string does not matched with the checksum calculated by the I-7530. For example: Error: t0012112209<CR> Right: t00121122FD<CR>
5	Timeout	The ASCII command strings are sent incomplete. For example: Error: T0018 Right: T0018<CR>

If the I-7530 CAN baud rate is not the same as the CAN baud rate on the CAN network, the data flow LED on the I-7530 will be flash with a constant frequency because the I-7530 cannot send any CAN messages to the CAN network. Therefore, users will need to read the I-7530 status by using the command "S[CHK]<CR>"(in the section 4.5) to help users understand what is going in the module. In general, the following errors could occur: CAN media connection problem, terminal resistor problem, different baud rate configuration with CAN network and the like.