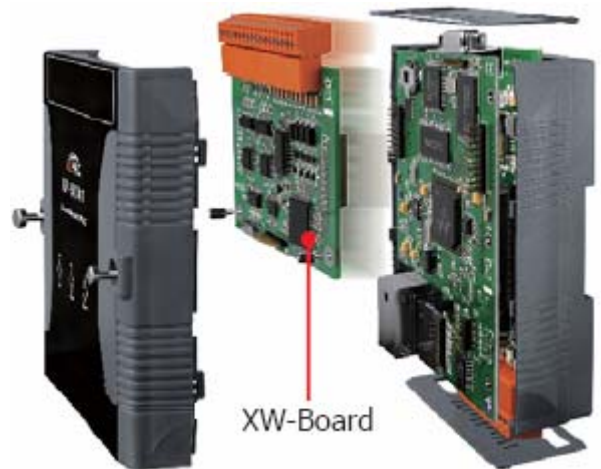


I/O Expansion Board

for LinPAC-51XX

User's Manual

Version 1.3, 2012/06/26



Warranty

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

Warning

ICP DAS assume no liability for damages consequent to 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, not for any infringements of patents or other rights of third parties resulting from its use.

Copyright

Copyright © 2011 by ICP DAS Co., Ltd. All rights are reserved.

Trademarks

Names are used for identification purposes and only may be registered trademarks of their respective companies

Table of Contents

1. Introduction.....	3
1.1 What's the I/O Expansion Bus?	3
1.2 Library-libxwboard.a	3
1.3 Demo program	3
1.4 i-Talk Utility	3
2. Wire Connection	4
2.1 Digital Input/Output Wire Connection.....	4
2.2 Voltage Input/Output Wire Connection.....	4
2.3 RS-232, RS-422, RS-485 Wire Connection.....	4
3. Expansion Boards.....	5
3.1 XW107: DI * 8 + DO * 8.....	5
3.2 XW107i: DI * 8 + DO * 8	10
3.3 XW110: DI * 16	11
3.4 XW304: AI * 6 + AO * 1 + DI * 4 + DO * 4.....	13
3.5 XW310: AI * 4 + AO * 2 + DI * 3 + DO * 3.....	24
3.6 XW506: RS-232 * 6.....	35
3.7 XW507: DI*5 + DO*5 + RS-422/485 * 1	36
3.8 XW508: RS-232 * 8.....	41
3.9 XW509: DI*4 + DO*4 + RS-232* 2	42
3.10 XW511i: RS-485 * 4	47
3.11 XW514: RS-485 * 8.....	48
4. Demo for XW-Boards	49
4.1 DI/O Expansion Boards	49
4.2 AI/O, DI/O Expansion Boards	52
4.3 RS-232/422/485, DI/O Expansion Boards	54

1. Introduction

1.1 What's the I/O Expansion Bus?

The LinPAC-51XXseries all support I/O expansion buses. The I/O expansion bus can be used to implement various I/O functions such as D/I, D/O, A/D, D/A, Timer/Counter, UART, flash memory, battery backup SRAM & other I/O functions. Nearly all kinds of I/O functions can be implemented on this bus.

Model	OS	CPU	Flash	SDRAM	Ethernet	VGA Resolution	USB	I/O Slot	Audio
LP-5131	Linux kernel 2.6	PXA270, 520 MHz	64 MB	128 MB	1	800 x 600	2	I/O expansion board optional	None
LP-5131-OD									Yes
LP-5141					2		1		None
LP-5141-OD									Yes

1.2 Library-libxwboard.a

In LinPAC-51XX SDK, ICP DAS provides the library file — **libxwboard.a** which includes all the functions from the I/O expansion buses which are used in the LinPAC-51XX Embedded Controller.

The **libxwboard.a** is designed specially for the I/O expansion buses on the Linux platform for use in the LinPAC-51XX which can be used to implement various I/O functions. Users can easily develop applications in the LinPAC-51XX by using either C or Java Language.

1.3 Demo program

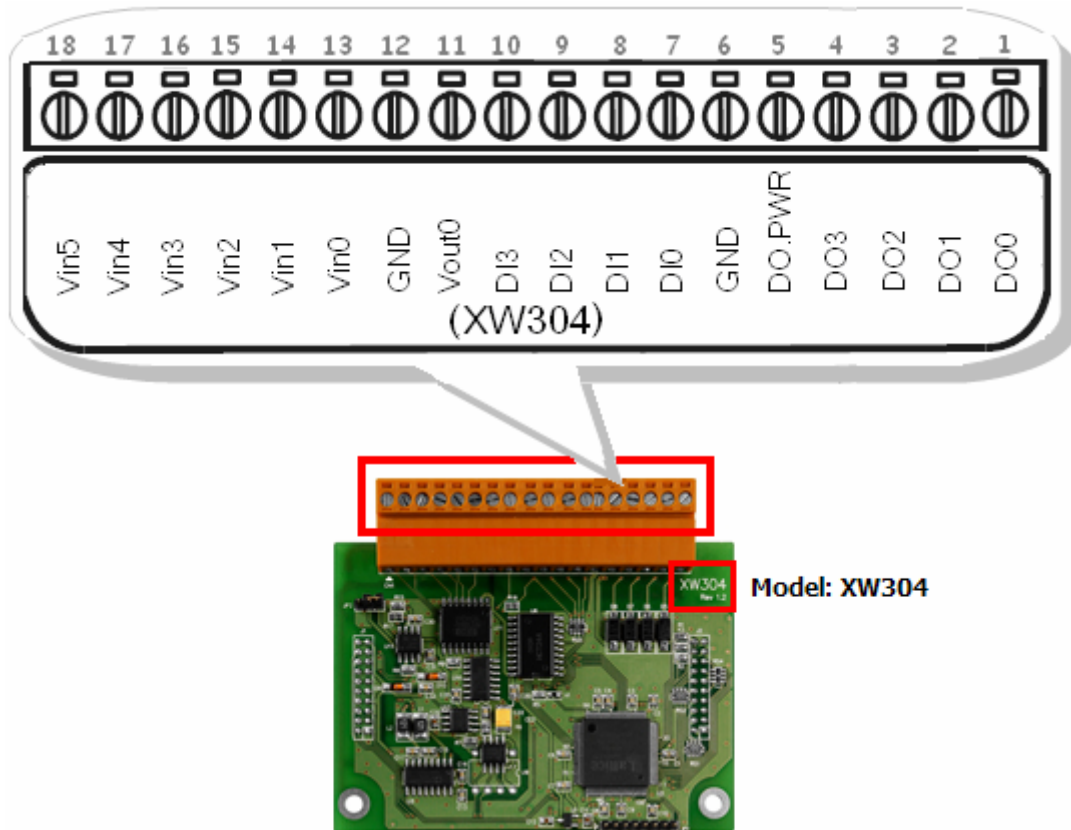
Download the demo programs of I/O expansion buses into LinPAC-51XX controller from LinPAC-51XX SDK, all of the demo programs are in <C:\cygwin\LinCon8k\examples\xwboard\>.

1.4 i-Talk Utility

LinPAC-51XX provide 'i-Talk utility' that make it convenient for users to access the modules, for more detail information please refer to:

ftp://ftp.icpdas.com/pub/cd/linpac/napdos/lp-5000/lp-51xx/lp-513x/user_manual/lp-5k_user_manual.pdf

2. Wire Connection



2.1 Digital Input/Output Wire Connection

Please refer to the website below:

http://www.icpdas.com.tw/product/solutions/pac/upac/xw107_hardware.html

2.2 Voltage Input/Output Wire Connection

Please refer to the website below:

http://www.icpdas.com.tw/product/solutions/pac/upac/xw304_hardware.html

2.3 RS-232, RS-422, RS-485 Wire Connection

Please refer to the website below:

http://www.icpdas.com.tw/product/solutions/pac/upac/xw506_hardware.html

3. Expansion Boards

3.1 XW107: DI * 8 + DO * 8

3.1.1 Specifications



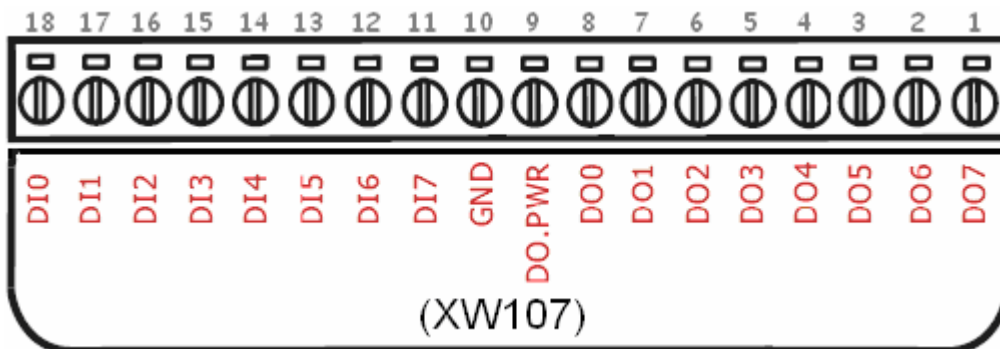
Digital Input:

- Channels: 8
- Input Type: Dry, Source
- Off Voltage Level: Open
- Off Voltage Level: Connect to GND
- Isolated: none
- Overvoltage Protection: 60 VDC

Digital Output:

- Channels: 8
- Output Type: Sink, Open Collector
- Output Voltage: +10 VCD ~ 40 VCD
- Max. Load Current: 200mA/channel at 25°C
- Isolated: none

3.1.2 Pin Assignment



Note: There is no need to use GND and DO.PWR in XW107(non-isolated).

3.1.3 Programming

➡ XW107_Init

Description:

This function is used to initialize the XW107.

Syntax:

```
int XW107_Init() [C]
```

Parameter:

None

Return Value:

On success, returns zero.

On error, returns -1, XW107 or XW107i cannot be found.

➡ XW107_Read_All_DI

Description:

This function is used to obtain all digital input value.

Syntax:

```
int XW107_Read_All_DI(void) [C]
```

Parameter:

None

Return Value:

data : 0x00~0xFF

1 : open

0 : close to GND

➡ XW107_Read_One_DI

Description:

This function is used to obtain each digital input value.

Syntax:

```
int XW107_Read_One_DI(int iChannel) [C]
```

Parameter:

iChannel : The digital input channel No.

Return Value:

1 : open

0 : close to GND

➡ XW107_Write_All_DO

Description:

This function is used to set the digital output value for all channel.

Syntax:

```
void XW107_Write_All_DO(int iOutValue) [C]
```

Parameter:

iOutValue: The digital output value. Range: 0x00~ 0xFF

Return Value:

None

➡ XW107_Write_One_DO**Description:**

This function is used to set the digital output value of the specific digital output channel No. of the XW107. The output value is only for “0” or “1”.

Syntax:

```
void XW107_Write_One_DO(int iChannel, int iStatus) [C]
```

Parameter:

iChannel : The digital output channel No.

iStatus =1 , Status is ON

iStatus =0 , Status is OFF

Return Value:

None

➡ XW107_Read_All_DO**Description:**

This function is used to obtain digital output readback All channels.

Syntax:

```
int XW107_Read_All_DO(void) [C]
```

Parameter:

None

Return Value:

0x00 ~ 0xFF

↻ XW107_Read_One_DO**Description:**

This function is used to obtain digital output readback one channels.

Syntax:

```
int XW107_Read_One_DO(int iChannel) [C]
```

Parameter:

iChannel : The digital output channel No.

Return Value:

1 : ON

0 : OFF

3.2 XW107i: DI * 8 + DO * 8

3.2.1 Specifications

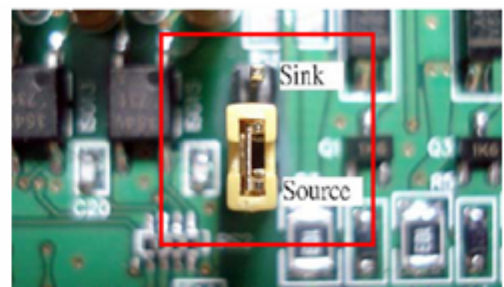
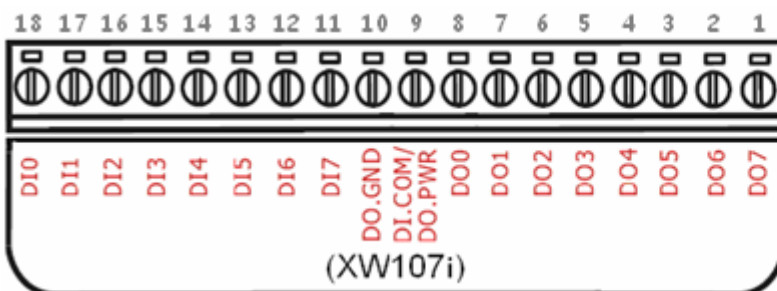
Digital Input:

- Channels: 8
- Input Type: Wet Contact (Sink/Source; Default)
- Off Voltage Level: +4 VCD max.
- On Voltage Level: +10 VCD ~50 VCD
- Input Impedance: 10K Ohm
- Overvoltage Protection: 60 VDC
- Intra-module Isolation, Field to Logic: 3750 Vrms

Digital Output:

- Channels: 8
- Output Type: Sink, Open Collector
- Output Voltage: +10 VCD ~40 VCD
- Max. Load Current: 200 mA / channel at 25°C
- Intra-module Isolation, Field to Logic: 3750 Vrms

3.2.2 Pin Assignment



Note: the GND and DO.PWR is only for XW107i (isolated).

3.2.3 Programming

Please refer to 3.1.3.

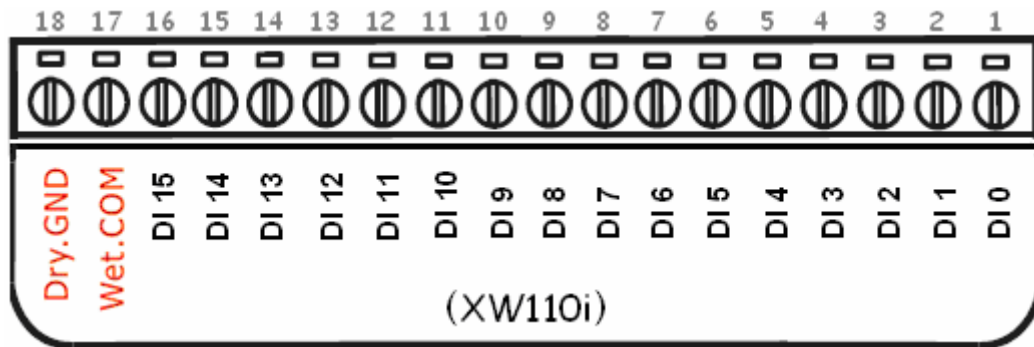
3.3 XW110: DI * 16

3.3.1 Specifications

Digital Input:

- ➔ Channels: 16
- ➔ Input Type: Dry + Wet Contact
- ➔ Dry contact (Off Voltage Level: Close to GND ; On Voltage Level: Close to GND)
- ➔ Wet contact (Off Voltage Level: Close to GND ; On Voltage Level: +10 VCD ~ +50 VCD)
- ➔ Overvoltage Protection: 60 VCD for wet contact
- ➔ Intra-module Isolation, Field to Logic: 3750 Vrms

3.4.2 Pin Assignment



3.3.3 Programming

➔ XW110_init

Description:

This function is used to initialize the XW110i.

Syntax:

```
int XW110_init() [C]
```

Parameter:

None

Return Value:

On success, returns zero.

On error, returns -1, XW110i cannot be found.

➡ XW110_Read_All_DI

Description:

This function is used to obtain all digital input value.

Syntax:

```
int XW110_Read_All_DI(void) [C]
```

Parameter:

None

Return Value:

data : 0x0000~0xffff

1 : open

0 : close to GND

➡ XW110_Read_One_DI

Description:

This function is used to obtain each digital input value.

Syntax:

```
int XW110_Read_One_DI(int iChannel) [C]
```

Parameter:

iChannel : The digital input channel No.

Return Value:

1 : open

0 : close to GND

3.4 XW304: AI * 6 + AO * 1 + DI * 4 + DO * 4

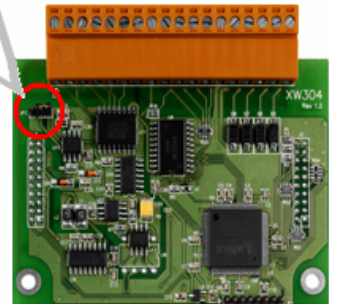
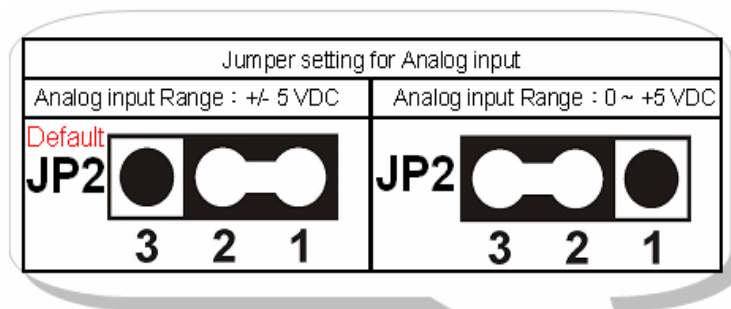
3.4 1 Specifications

Analog Input:

- Channel: 6
- Wiring: Single-Ended
- Input Impedance: 1 MOhm
- Range: +/- 5 VDC, 0 ~ +5 VDC (Default: +/- 5 VDC)
- Overvoltage Protection: +/- 30 VDC
- Sampling Rate: 4 KHz
- Resolution: 12-bit
- Isolation: none

Analog Output:

- Channel: 1
- Wiring: Single-Ended
- Resolution: 12-bit
- Range: +/- 5 VDC
- Isolation: none



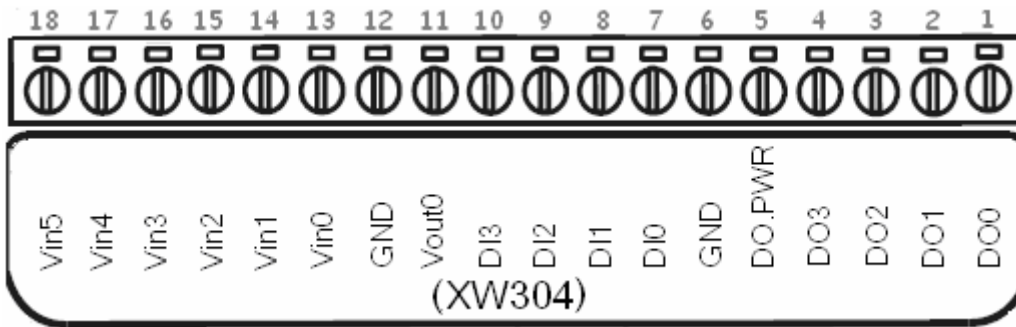
Digital Input:

- Channel: 4
- Contact: Dry
- Dry Contact: On Voltage Level → Close to GND
Off Voltage Level → Open
- Overvoltage Protection: 30 VDC
- Output Capacity: 20 mA

Digital Output:

- Channel: 4
- Type: Open Collector
- Sink/Source (NPN/PNP): Sink
- Load Voltage: +10 VDC ~ 40 VDC
- Max. Load Current: 200 mA/channel at 25 °C
- Overload Protection: 1.4 A

3.4.2 Pin Assignment



3.4.3 Programming

➡ XW304_init

Description:

This function is used to initialize the XW304.

Syntax:

```
int XW304_init(void) [C]
```

Parameter:

None

Return Value:

0: success

Non-zero: Bit0: 1 => Read A/D Gain failure (+/-5V)

Bit1: 1 => Read A/D Offset failure (+/-5V)

Bit2: 1 => Read A/D Gain failure (0~5V)

Bit3: 1 => Read A/D Offset failure (0~5V)

Bit4: 1 => Read D/A Gain failure (+5V)

Bit5: 1 => Read D/A Offset failure (+5V)

Bit7: 1 => XW304 cannot be found

➤ XW304_GetLibVersion

Description:

This function is used to get the version number of XW304 library.

Syntax:

```
unsigned XW304_GetLibVersion(void) [C]
```

Parameter:

None

Return Value:

It always returns a non-zero value to indicate the current version.

For example, return value= 0x100, it means the version is 1.00

➤ XW304_Read_AD_Gain

Description:

This function is used to read the calibration data stored in the EEPROM.

Syntax:

```
float XW304_Read_AD_Gain(int ai_type) [C]
```

Parameter:

a_type : Analog input range selectable, 0 for (+/-5V) and 1 for (0~5V)

This argument must correspond with the jumper-selectable setting on the XW304

Return Value:

On error, returns 10.0 if there was no setting in the EEPROM

↻ XW304_Read_AD_Offset**Description:**

This function is used to read the calibration data stored in the EEPROM.

Syntax:

```
float XW304_Read_AD_Offset(int ai_type) [C]
```

Parameter:

a_type : Analog input range selectable, 0 for (+/-5V) and 1 for (0~5V)

This argument must correspond with the jumper-selectable setting on the XW304

Return Value:

On error, returns 10.0 if there was no setting in the EEPROM

↻ XW304_Read_DA_Gain**Description:**

This function is used to read the calibration data stored in the EEPROM.

Syntax:

```
float XW304_Read_DA_Gain(void) [C]
```

Parameter:

None

Return Value:

On error, returns 10.0 if there was no setting in the EEPROM

↻ XW304_Read_DA_Offset

Description:

This function is used to read the calibration data stored in the EEPROM.

Syntax:

```
float XW304_Read_DA_Offset(void) [C]
```

Parameter:

None

Return Value:

On error, returns 10.0 if there was no setting in the EEPROM

↻ XW304_SetDelay

Description:

This function is used specify a short time delay, in milliseconds, to wait before obtaining the raw data from the analog to digital converter chip.

Syntax:

```
void XW304_SetDelay(unsigned delay) [C]
```

Parameter:

delay: Delay time, represented as a integer in the range [0, 1000] (unit: 0.01 ms)

Return Value:

None

➤ XW304_GetDelay

Description:

This function is used to read back the time delay.

Syntax:

```
unsigned XW304_GetDelay(void) [C]
```

Parameter:

None

Return Value:

time delay

➤ XW304_AnalogIn

Description:

This function is used to read the current input voltage, represented as a float in the range [-5, 5] or [0, 5].

Syntax:

```
float XW304_AnalogIn(int ai_type, int channel) [C]
```

Parameter:

ai_type: Analog input range selectable, 0 for (+/-5V) and 1 for (0~5V)

This argument must correspond with the jumper-selectable setting on the XW304 channel: Channel number, represented as 0, 1, 2, 3, 4 and 5

Return Value:

Return the voltage supplied by a voltage source.

➤ XW304_AnalogInSetChannel

Description:

This function is used to specify the input channel number before the call of XW304_AnalogInHex().

Syntax:

```
void XW304_AnalogInSetChannel(int channel) [C]
```

Parameter:

channel: Channel number, represented as 0, 1, 2, 3, 4 and 5.

Return Value:

None

➤ XW304_AnalogInHex

Description:

This function is used to read a raw data from the analog to digital converter chip.

Syntax:

```
int XW304_AnalogInHex(void) [C]
```

Parameter:

None

Return Value:

A raw data obtained from the analog to digital converter chip, represented as a hex value in the range [0, 0xFFFF]

➤ XW304_AnalogInHexToFloat

Description:

This function is used to convert a raw data reading into a voltage reading.

Syntax:

```
float XW304_AnalogInHexToFloat(int ai_type, int value) [C]
```

Parameter:

ai_type: Analog input range selectable, 0 for [-5, 5] and 1 for [0, 5]

This argument must correspond with the jumper-selectable setting on the XW304

value: The raw data returned by XW304_AnalogInHex()

Return Value:

Return the voltage, represented as a float.

➤ XW304_AnalogOut

Description:

This function is used to convert a raw data reading into a voltage reading.

Syntax:

```
void XW304_AnalogOut(float value) [C]
```

Parameter:

value: Set the output voltage, represented as a float in the range [-5, 5]

Return Value:

None

↻ XW304_Read_All_DI

Description:

This function is used to read the input status of all channels.

Syntax:

```
int XW304_Read_All_DI(void) [C]
```

Parameter:

None

Return Value:

All digital inputs are packed as one channel per bit of the return data. Status is indicated as 0=ON (short to GND) and 1=OFF (Open)

For example:

The status of inputs is shown as the byte value 0C hex, or binary 0000 1010. DI3 is in the fifth bit position from the left, and DI0 is the LSB of this byte. The four remaining high order bits are zero filled.

↻ XW304_Read_One_DI

Description:

This function is used to read the input status of the specified channel.

Syntax:

```
void XW304_Read_One_DI(int channel) [C]
```

Parameter:

channel: Channel number, represented as 0, 1, 2 and 3

Return Value:

0: ON (short to GND)

1: OFF (Open)

➤ XW304_Write_All_DO**Description:**

This function is used to set the output status for all channels.

Syntax:

```
void XW304_Write_All_DO(int value) [C]
```

Parameter:

value: All digital outputs are packed as one channel per bit of the writing value

Return Value:

None

➤ XW304_Write_One_DO**Description:**

This function is used to set the output status for the specified channel.

Syntax:

```
void XW304_Write_One_DO(int channel, int status) [C]
```

Parameter:

channel: Channel number, represented as 0, 1, 2 and 3

status: An integer specifying the output value, 0 for OFF and 1 for ON

Return Value:

None

↻ XW304_Read_All_DO

Description:

This function is used to read back the current status of all channels

Syntax:

```
int XW304_Read_All_DO(void) [C]
```

Parameter:

None

Return Value:

All digital outputs are packed as one channel per bit of the return data. Status is indicated as 1=ON and 0=OFF

For example:

The status of outputs is shown as the byte value 0C hex, or binary 0000 1010. DO3 is in the fifth bit position from the left, and DO0 is the LSB of this byte. The four remaining high order bits are zero filled.

↻ XW304_Read_One_DO

Description:

This function is used to read back the current status of the specified channel.

Syntax:

```
int XW304_Read_One_DO(int channel) [C]
```

Parameter:

channel: Channel number, represented as 0, 1, 2 and 3

Return Value:

0: OFF

1: ON

3.5 XW310: AI * 4 + AO * 2 + DI * 3 + DO * 3

~Available soon~

3.5.1 Specifications

Analog Input:

- Channel: 4
- Wiring: Differentail
- Input Impedance: 1 MOhm
- Range: +/- 10 VDC
- Overvoltage Protection: +/- 30 VDC
- Sampling Rate: 4 KHz
- Resolution: 12-bit
- Isolation: none

Analog Output:

- Channel: 2
- Wiring: Single-Ended
- Resolution: 12-bit
- Range: +/- 10 VDC
- Isolation: none

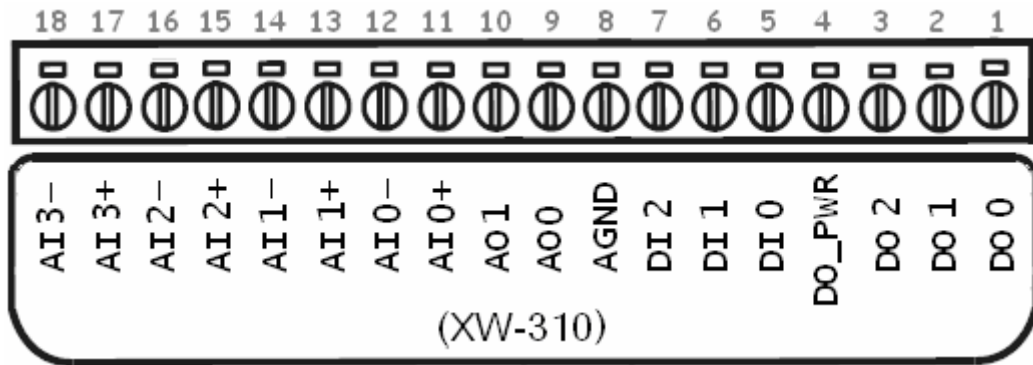
Digital Input:

- Channel: 3
- Contact: Dry
- Dry Contact: On Voltage Level → Close to GND
Off Voltage Level → Open
- Overvoltage Protection: 30 VDC
- Output Capacity: 30 mA

Digital Output:

- Channel: 3
- Type: Open Collector
- Sink/Source (NPN/PNP): Sink
- Load Voltage: +10 VDC ~ 40 VDC
- Max. Load Current: 200 mA/channel at 25 °C
- Overload Protection: 1.4 A

3.5.2 Pin Assignment



3.5.3 Programming

➔ XW310_Init

Description:

This function is used to initialize the XW310.

Syntax:

```
int XW310_Init(void) [C]
```

Parameter:

None

Return Value:

- 0: success
- Non-zero: Bit0: 1 => Read A/D Gain failure
- Bit1: 1 => Read A/D Offset failure
- Bit2: 1 => Read A/D Gain failure
- Bit3: 1 => Read A/D Offset failure
- Bit4: 1 => Read D/A Gain failure
- Bit5: 1 => Read D/A Offset failure
- Bit7: 1 => XW310 cannot be found

↻ XW310_GetLibVersion

Description:

This function is used to get the version number of XW310 library.

Syntax:

```
unsigned XW310_GetLibVersion(void) [C]
```

Parameter:

None

Return Value:

It always returns a non-zero value to indicate the current version.

For example, return value= 0x100, it means the version is 1.00

↻ XW310_Read_AD_Gain

Description:

This function is used to read the calibration data stored in the EEPROM.

Syntax:

```
float XW310_Read_AD_Gain() [C]
```

Parameter:

a_type : Analog input range selectable

This argument must correspond with the jumper-selectable setting on the XW310

Return Value:

On error, returns 10.0 if there was no setting in the EEPROM

↻ XW310_Read_AD_Offset

Description:

This function is used to read the calibration data stored in the EEPROM.

Syntax:

```
float XW310_Read_AD_Offset [C]
```

Parameter:

a_type : Analog input range selectable,

This argument must correspond with the jumper-selectable setting on the XW310

Return Value:

On error, returns 10.0 if there was no setting in the EEPROM

↻ XW310_Read_DA_Gain

Description:

This function is used to read the calibration data stored in the EEPROM.

Syntax:

```
float XW310_Read_DA_Gain(void) [C]
```

Parameter:

None

Return Value:

On error, returns 10.0 if there was no setting in the EEPROM

↻ XW310_Read_DA_Offset

Description:

This function is used to read the calibration data stored in the EEPROM.

Syntax:

```
float XW310_Read_DA_Offset(void) [C]
```

Parameter:

None

Return Value:

On error, returns 10.0 if there was no setting in the EEPROM

↻ XW310_SetDelay

Description:

This function is used specify a short time delay, in milliseconds, to wait before obtaining the raw data from the analog to digital converter chip.

Syntax:

```
void XW310_SetDelay(unsigned delay) [C]
```

Parameter:

delay: Delay time, represented as a integer in the range [0, 1000] (unit: 0.01 ms)

Return Value:

None

↻ XW310_GetDelay

Description:

This function is used to read back the time delay.

Syntax:

```
unsigned XW310_GetDelay(void) [C]
```

Parameter:

None

Return Value:

time delay

↻ XW310_AnalogIn

Description:

This function is used to read the current input voltage, represented as a float.

Syntax:

```
float XW310_AnalogIn(int channel) [C]
```

Parameter:

channel: Channel number, represented as 0, 1, 2 and 3

Return Value:

Return the voltage supplied by a voltage source.

➤ XW310_AnalogInSetChannel

Description:

This function is used to specify the input channel number before the call of XW310_AnalogInHex().

Syntax:

```
void XW310_AnalogInSetChannel(int channel) [C]
```

Parameter:

channel: Channel number, represented as 0, 1, 2 and 3

Return Value:

None

➤ XW310_AnalogInHex

Description:

This function is used to read a raw data from the analog to digital converter chip.

Syntax:

```
int XW310_AnalogInHex(void) [C]
```

Parameter:

None

Return Value:

A raw data obtained from the analog to digital converter chip, represented as a hex value in the range [0, 0xFFFF]

➡ XW310_AnalogInHexToFloat

Description:

This function is used to convert a raw data reading into a voltage reading.

Syntax:

```
float XW310_AnalogInHexToFloat(int value) [C]
```

Parameter:

value: The raw data returned by XW310_AnalogInHex()

Return Value:

Return the voltage, represented as a float.

➡ XW310_AnalogOut

Description:

This function is used to convert a raw data reading into a voltage reading.

Syntax:

```
void XW310_AnalogOut(int channel, float value) [C]
```

Parameter:

channel: Channel number, represented as 0 and 1

value: Set the output voltage, represented as a float in the range [-10, 10]

Return Value:

None

↻ XW310_Read_All_DI

Description:

This function is used to read the input status of all channels.

Syntax:

```
int XW310_Read_All_DI(void) [C]
```

Parameter:

None

Return Value:

All digital inputs are packed as one channel per bit of the return data. Status is indicated as 0=ON (short to GND) and 1=OFF (Open)

For example:

The status of inputs is shown as the byte value 0C hex, or binary 0000 1010. DI2 is in the eight bit position from the left, and DI0 is the LSB of this byte. The four remaining high order bits are zero filled.

↻ XW310_Read_One_DI

Description:

This function is used to read the input status of the specified channel.

Syntax:

```
int XW310_Read_One_DI(int channel) [C]
```

Parameter:

channel: Channel number, represented as 0, 1 and 2

Return Value:

0: ON (short to GND)

1: OFF (Open)

↻ XW310_Write_All_DO**Description:**

This function is used to set the output status for all channels.

Syntax:

```
void XW310_Write_All_DO(int value) [C]
```

Parameter:

value: All digital outputs are packed as one channel per bit of the writing value

Return Value:

None

↻ XW310_Write_One_DO**Description:**

This function is used to set the output status for the specified channel.

Syntax:

```
void XW310_Write_One_DO(int channel, int status) [C]
```

Parameter:

channel: Channel number, represented as 0, 1 and 2

status: An integer specifying the output value, 0 for OFF and 1 for ON

Return Value:

None

↻ XW310_Read_All_DO

Description:

This function is used to read back the current status of all channels

Syntax:

```
int XW310_Read_All_DO(void) [C]
```

Parameter:

None

Return Value:

All digital outputs are packed as one channel per bit of the return data. Status is indicated as 1=ON and 0=OFF

For example:

The status of outputs is shown as the byte value 0C hex, or binary 0000 1010. DO2 is in the eight bit position from the left, and DO0 is the LSB of this byte. The four remaining high order bits are zero filled.

↻ XW310_Read_One_DO

Description:

This function is used to read back the current status of the specified channel.

Syntax:

```
int XW310_Read_One_DO(int channel) [C]
```

Parameter:

channel: Channel number, represented as 0, 1 and 2

Return Value:

0: OFF

1: ON

3.6 XW506: RS-232 * 6

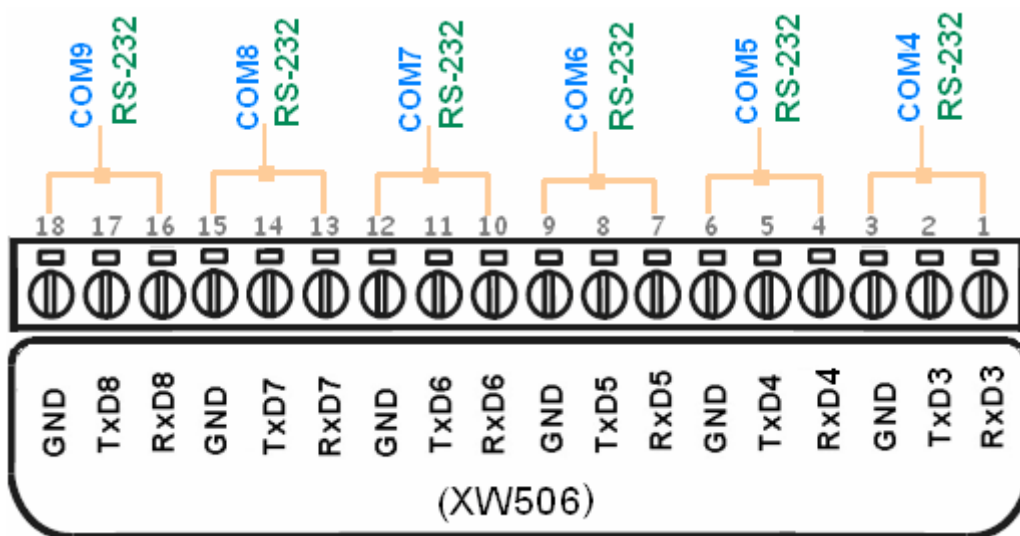
~Available soon~

3.7.1 Specifications

Parallel I/O:

- ➔ RS-232 *6
- ➔ 16954 compatible
- ➔ Internal FIFO: 16 bytes
- ➔ Transmission speed: 1152.K BPS max.
- ➔ Isolated: None

3.5.2 Pin Assignment



COM port	Definitions in LP-5K SDK	Device name	Default baud rate
4	COM4	ttyS2	9600
5	COM5	ttyS3	9600
6	COM6	ttyS4	9600
7	COM7	ttyS5	9600
8	COM8	ttyS6	9600
9	COM9	ttyS7	9600

3.7 XW507: DI*5 + DO*5 + RS-422/485 * 1

~Available soon~

3.7.1 Specifications

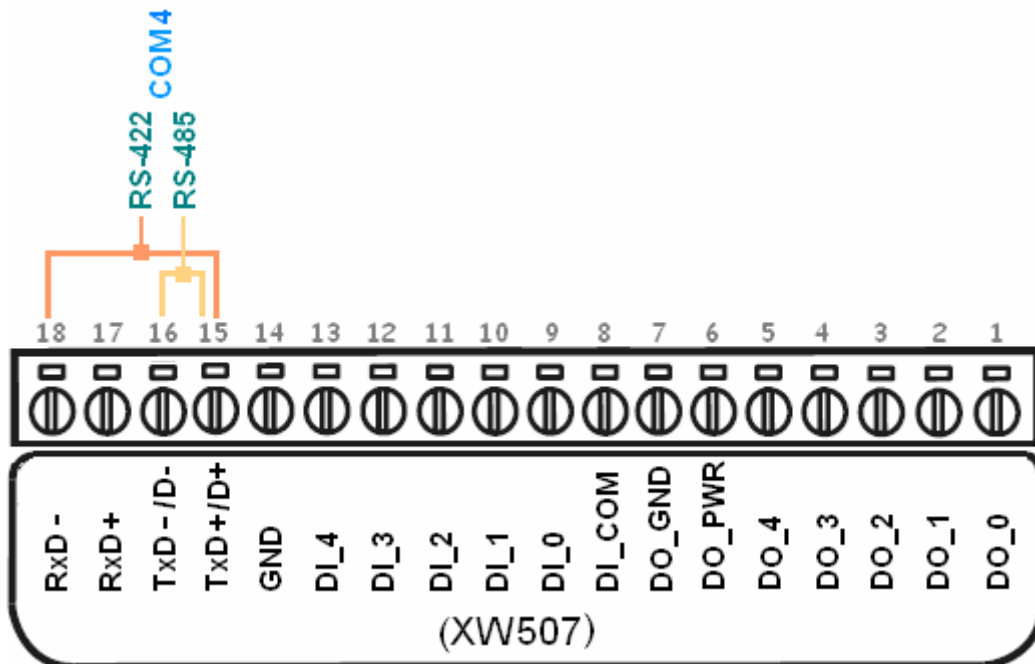
Digital I/O:

- ➔ DI * 5
Input Range/ Type: Logic high level (3.5V~30V) / Logic low level (0V~1V)
- ➔ DO *5
Open-collector Output: 100 mA / 30V
- ➔ Isolated: none

Parallel I/O:

- ➔ RS-422/485 *1
- ➔ 16954 compatible
- ➔ Internal FIFO: 16 bytes
- ➔ Transmission speed: 1152.K BPS max.
- ➔ Isolated: None

3.7.2 Pin Assignment



COM port	Definitions in LP-5K SDK	Device name	Default baudrate
4	COM4	ttyS2	9600

3.7.3 Programming

➤ XW507_Init

Description:

This function is used to initialize the XW-507.

Syntax:

```
int XW507_Init() [C]
```

Parameter:

None

➤ XW507_Read_All_DI

Description:

This function is used to obtain all digital input value.

Syntax:

```
int XW507_Read_All_DI(void) [C]
```

Parameter:

None

Return Value:

data : 0x0000~0xffff

1 : open

0 : close to GND

↻ XW507_Read_One_DI

Description:

This function is used to obtain each digital input value.

Syntax:

```
int XW507_Read_One_DI(int iChannel) [C]
```

Parameter:

iChannel : The digital input channel No.

Return Value:

1 : open

0 : close to GND

↻ XW507_Write_All_DO

Description:

This function is used to set the digital output value for all channel.

Syntax:

```
void XW507_Write_All_DO(int iOutValue) [C]
```

Parameter:

iOutValue: The digital output value. Range: 0x00~ 0xFF

Return Value:

None

➡ XW507_Write_One_DO**Description:**

This function is used to set the digital output value of the specific digital output channel No. of the XW-507. The output value is only for "0" or "1".

Syntax:

```
void XW507_Write_One_DO(int iChannel, int iStatus) [C]
```

Parameter:

iChannel : The digital output channel No.

iStatus =1 , Status is ON

iStatus =0 , Status is OFF

Return Value:

None

➡ XW507_Read_All_DO**Description:**

This function is used to obtain digital output readback All channels.

Syntax:

```
int XW507_Read_All_DO(void) [C]
```

Parameter:

None

Return Value:

0x00 ~ 0xFF

↻ XW507_Read_One_DO**Description:**

This function is used to obtain digital output readback one channels.

Syntax:

```
int XW507_Read_One_DO(int iChannel) [C]
```

Parameter:

iChannel : The digital output channel No.

Return Value:

1 : ON

0 : OFF

3.8 XW508: RS-232 * 8

~Available soon~

3.8.1 Specifications

Parallel I/O:

- ➔ RS-232 *8
- ➔ 16954 compatible
- ➔ Internal FIFO: 16 bytes
- ➔ Transmission speed: 1152.K BPS max.
- ➔ Isolated: None

3.8.2 Pin Assignment

3.9 XW509: DI*4 + DO*4 + RS-232* 2

~Available soon~

3.9.1 Specifications

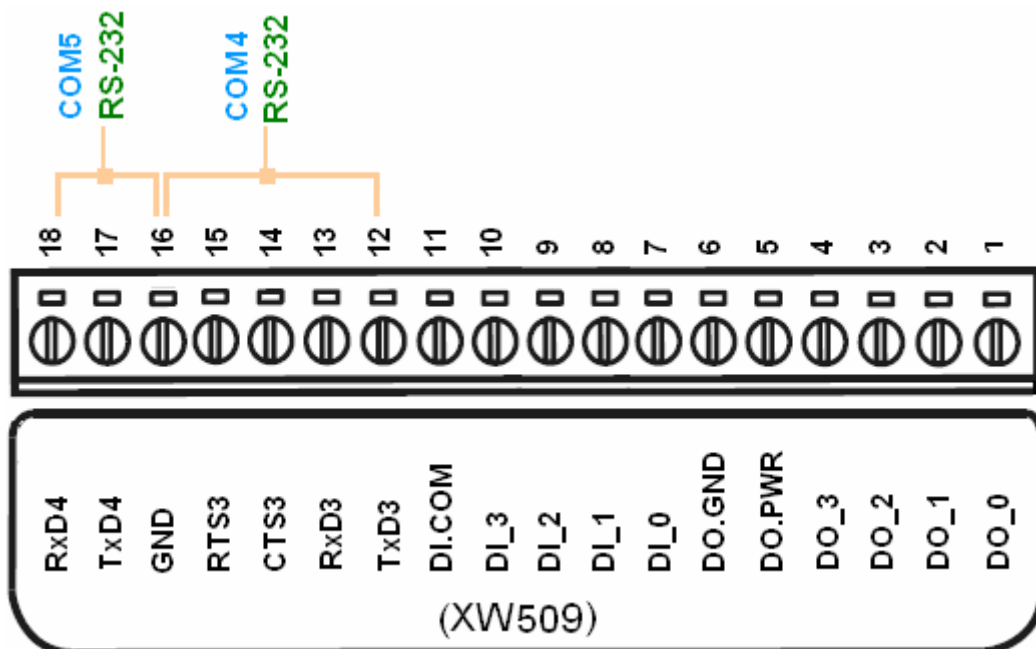
Digital I/O:

- DI * 4
Input Range/ Type: Logic high level (3.5V~30V) / Logic low level (0V~1V)
- DI * 4
Open-collector Output: 100 mA / 30V
- Isolated: none

Parallel I/O:

- RS-232 *2
- 16954 compatible
- Internal FIFO: 16 bytes
- Transmission speed: 1152.K BPS max.
- Isolated: None

3.9.2 Pin Assignment



COM port	Definitions in LP-5K SDK	Device name	Default baudrate
4	COM4	ttyS2	9600
5	COM5	ttyS3	9600

3.9.3 Programming

➤ XW509_Init

Description:

This function is used to initialize the XW509.

Syntax:

```
int XW509_init() [C]
```

Parameter:

None

➤ XW509_Read_All_DI

Description:

This function is used to obtain all digital input value.

Syntax:

```
int XW509_Read_All_DI(void) [C]
```

Parameter:

None

Return Value:

data : 0x0000~0xffff

1 : open

0 : close to GND

↻ XW509_Read_One_DI

Description:

This function is used to obtain each digital input value.

Syntax:

```
int XW509_Read_One_DI(int iChannel) [C]
```

Parameter:

iChannel : The digital input channel No.

Return Value:

1 : open

0 : close to GND

↻ XW509_Write_All_DO

Description:

This function is used to set the digital output value for all channel.

Syntax:

```
void XW509_Write_All_DO(int iOutValue) [C]
```

Parameter:

iOutValue: The digital output value. Range: 0x00~ 0xFF

Return Value:

None

➡ XW509_Write_One_DO

Description:

This function is used to set the digital output value of the specific digital output channel No. of the XW509. The output value is only for “0” or “1”.

Syntax:

```
[C]  
void XW509_Write_One_DO(int iChannel, int iStatus)
```

Parameter:

iChannel : The digital output channel No.

iStatus =1 , Status is ON

iStatus =0 , Status is OFF

Return Value:

None

➡ XW509_Read_All_DO

Description:

This function is used to obtain digital output readback All channels.

Syntax:

```
[C]  
int XW509_Read_All_DO(void)
```

Parameter:

None

Return Value:

0x00 ~ 0xFF

↻ XW509_Read_One_DO**Description:**

This function is used to obtain digital output readback one channels.

Syntax:

```
int XW509_Read_One_DO(int iChannel) [C]
```

Parameter:

iChannel : The digital output channel No.

Return Value:

1 : ON

0 : OFF

3.10 XW511i: RS-485 * 4

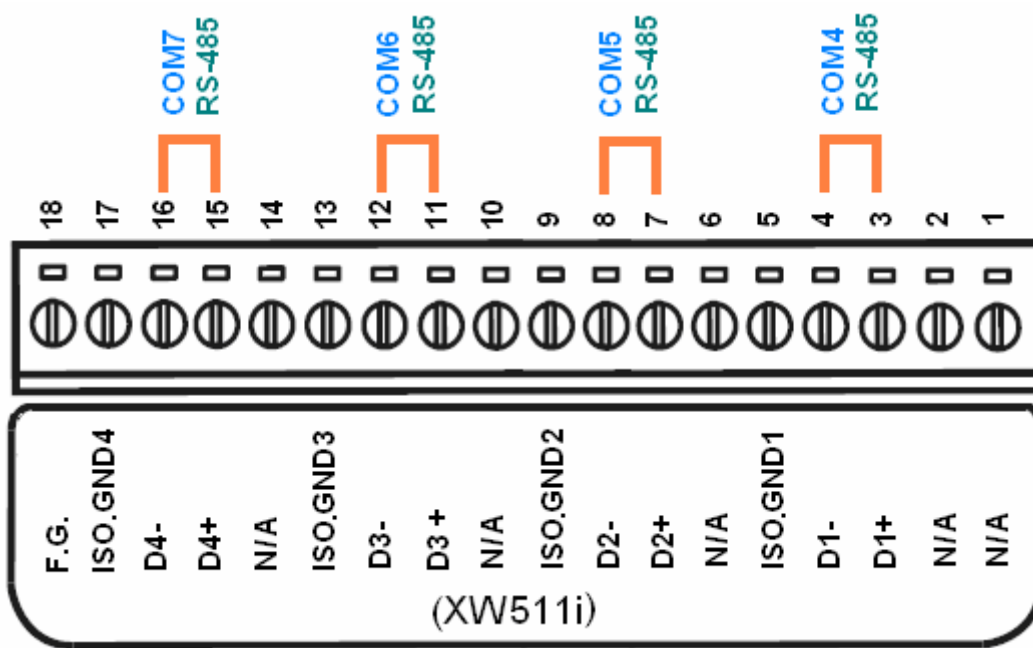
~Available soon~

3.10.1 Specifications

Parallel I/O:

- RS-485 *4
- 16954 compatible
- Internal FIFO: 16 bytes
- Transmission speed: 1152.K BPS max.
- Isolated: Yes

3.10.2 Pin Assignment



COM port	Definitions in LP-5K SDK	Device name	Default baudrate
4	COM4	ttyS2	9600
5	COM5	ttyS3	9600
6	COM6	ttyS4	9600
7	COM7	ttyS5	9600

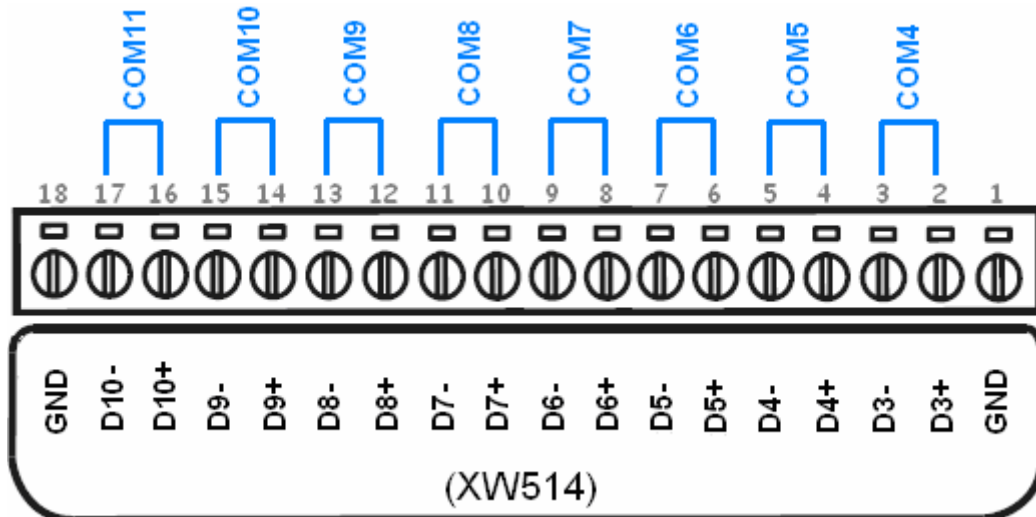
3.11 XW514: RS-485 * 8

3.11.1 Specifications

Parallel I/O:

- RS-485 *8
- 16954 compatible
- Internal FIFO: 16 bytes
- Transmission speed: 1152.K BPS max.
- Isolated: None

3.11.2 Pin Assignment



COM port	Definitions in LP-5K SDK	Device name	Default baudrate
4	COM4	ttyS2	9600
5	COM5	ttyS3	9600
6	COM6	ttyS4	9600
7	COM7	ttyS5	9600
8	COM8	ttyS6	9600
9	COM9	ttyS7	9600
10	COM10	ttyS8	9600
11	COM11	ttyS9	9600

4. Demo for XW-Boards

4.1 DI/O Expansion Boards

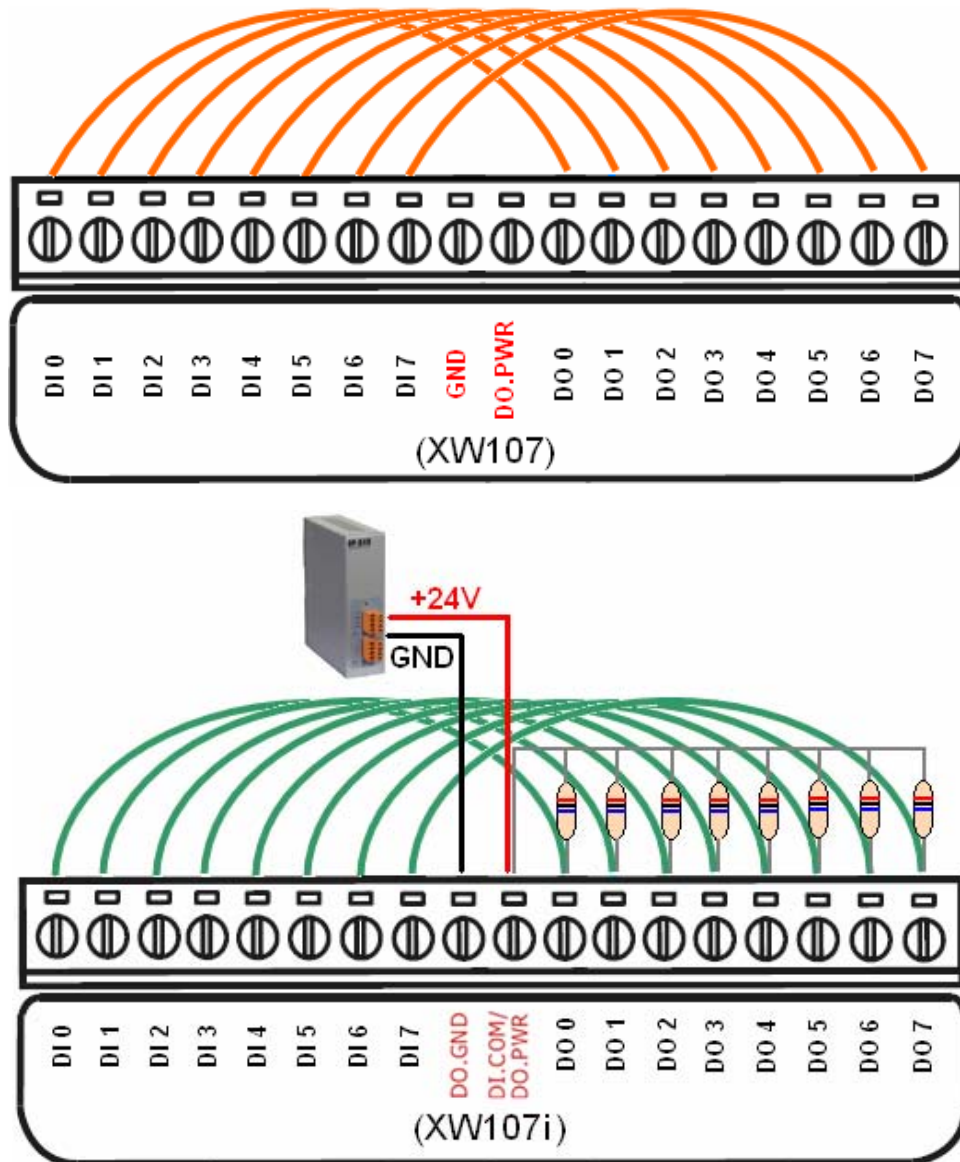
4.1.1 DIO

📍 Location

To install LinPAC-51XX SDK first from LinPAC-51XX's CD or FTP site of ICP DAS, and user can refer to C:\cygwin\LinCon8k\examples\xwboard\xw1xx\xw107.c

🔌 Wire Connection

Connect to DI and DO as below:



Note: There is no need to use GND and DO.PWR in XW107(non-isolated), the GND and DO.PWR is only for XW107I (isolated).

 Run

Step1: Download xw107.exe in LinPAC-51XX.

Step2: Changes the permission of a file as below:

```
# chmod 755 xw107.exe
```

Step3: Running program.

```
# ./xw107.exe
```

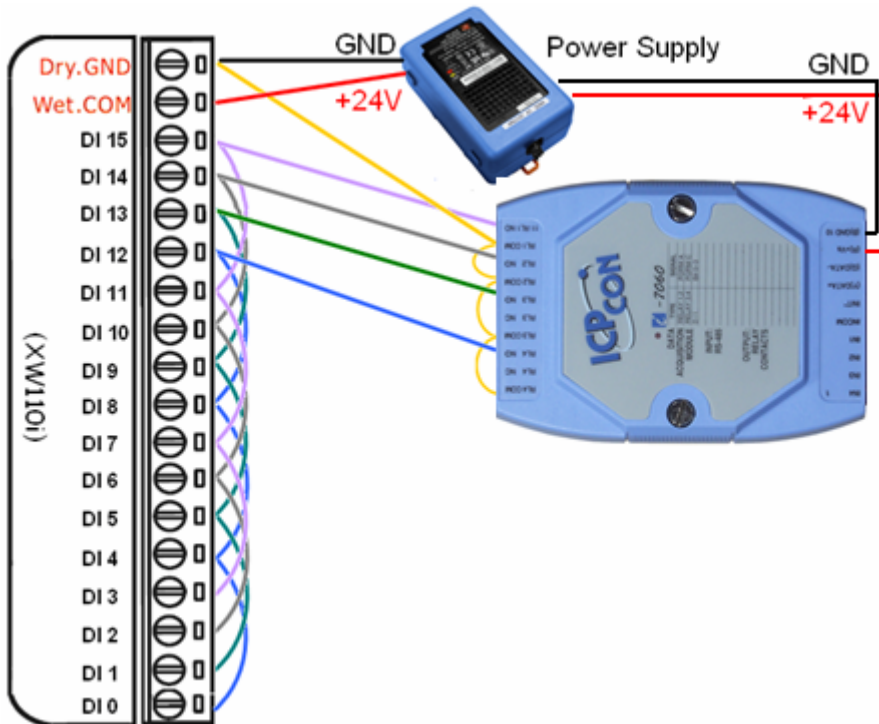
4.1.2 DI

➡ Location

To install LinPAC-51XX SDK first from LinPAC-51XX's CD or FTP site of ICP DAS, and user can refer to C:\cygwin\LinCon8k\examples\xwboard\xw-1xx\xw110.c

➡ Wire Connection

Connect to DI as below:



➡ Run

Step1: Download xw110.exe in LinPAC-51XX.

Step2: Changes the permission of a file as below:

```
# chmod 755 xw110.exe
```

Step3: Running program.

```
# ./xw110.exe
```

```
10.1.0.34 [81x44]
# ./xw110.exe
Demo program for LinPAC-5000 + XW110i

Please choose one of the following :
0.) Exit
1.) Read All DI channels.
2.) Read One DI channel.

Choose an option and press [Enter]:1
Input value=> 0xffff
#
```

```
10.1.0.34 [81x44]
# ./xw110.exe
Demo program for LinPAC-5000 + XW110i

Please choose one of the following :
0.) Exit
1.) Read All DI channels.
2.) Read One DI channel.

Choose an option and press [Enter]:2
Please select DI channel(0~15): 3
Channel 3 ==> ON
#
```

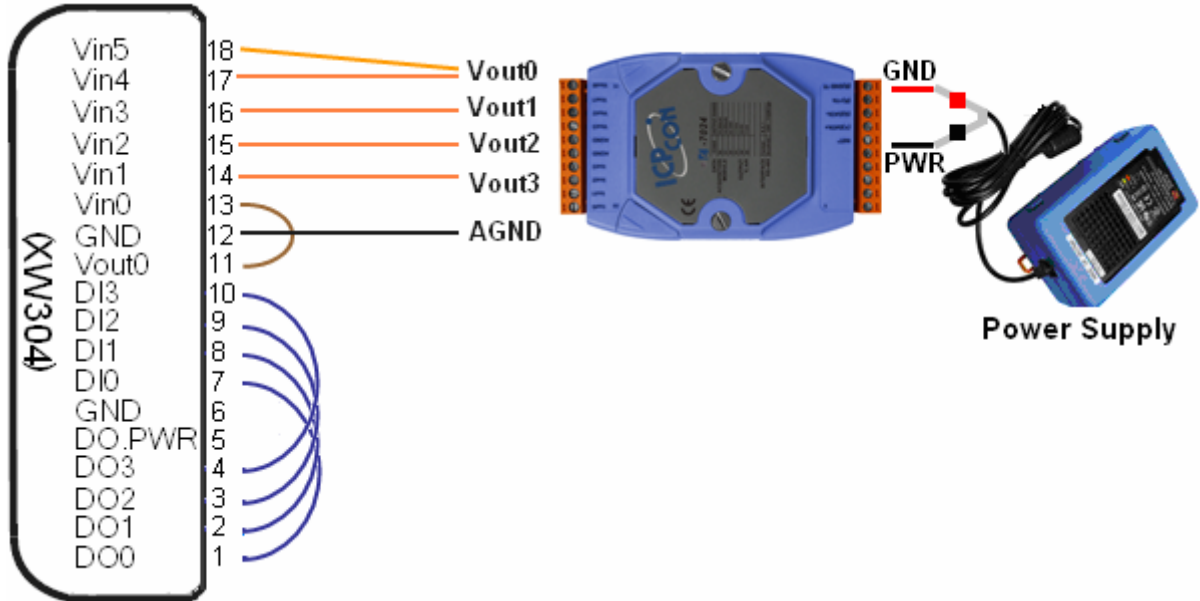
4.2 AI/O, DI/O Expansion Boards

➔ Location

To install LinPAC-51XX SDK first from LinPAC-51XX's CD or FTP site of ICP DAS, and user can refer to C:\cygwin\LinCon8k\examples\xwboard\xw3xx\xw304.c

➔ Wire Connection

Connect to DI and DO as below:



➔ Run

Step1: Download xw304.exe in LinPAC-51XX.

Step2: Changes the permission of a file as below:

```
# chmod 755 xw304.exe
```

Step3: Running program.

```
# ./xw304.exe
```

```

Demo program for LinPAC-5000-XW304
-----
Please choose one of the following :
 0.) Read the calibration data
 1.) Analog Input(+/-5V)
 2.) Analog Input(0~5V)
 3.) Analog Output(0~5V)
 4.) Read All DI channels
 5.) Read One DI channel
 6.) Write All DO channels
 7.) Write One DO channel
 8.) DO Readback All channels
 9.) DO Readback One channel
10.) Exit
Note1:AI(Ch0~5);AO(Ch0);DO(Ch0~3);DI(Ch0~3)
-----
Choose an option and press [Enter]: 0
EEPROM A/D Gain  => [0.002478]
EEPROM A/D Offset => [-5.069600]
EEPROM A/D Gain  => [0.002479]
EEPROM A/D Offset => [-5.073890]
EEPROM D/A Gain  => [0.002543]
EEPROM D/A Offset => [-5.204710]

```

DI	AI
Demo program for LinPAC-5000-XW304	Demo program for LinPAC-5000-XW304
Please choose one of the following : 0.) Read the calibration data 1.) Analog Input(+/-5V) 2.) Analog Input(0~5V) 3.) Analog Output(0~5V) 4.) Read All DI channels 5.) Read One DI channel 6.) Write All DO channels 7.) Write One DO channel 8.) DO Readback All channels 9.) DO Readback One channel 10.) Exit Note1:AI(Ch0~5);AO(Ch0);DO(Ch0~3);DI(Ch0~3)	Please choose one of the following : 0.) Read the calibration data 1.) Analog Input(+/-5V) 2.) Analog Input(0~5V) 3.) Analog Output(0~5V) 4.) Read All DI channels 5.) Read One DI channel 6.) Write All DO channels 7.) Write One DO channel 8.) DO Readback All channels 9.) DO Readback One channel 10.) Exit Note1:AI(Ch0~5);AO(Ch0);DO(Ch0~3);DI(Ch0~3)
Choose an option and press [Enter]: 7	Choose an option and press [Enter]: 3
DO Channel (0~3): 0 DO status (0:OFF, 1:ON): 1 DOO => ON	Voltage (-5.0 ~ +5.0): 4 AO => 4.0000V
DO	AO
Demo program for LinPAC-5000-XW304	Demo program for LinPAC-5000-XW304
Please choose one of the following : 0.) Read the calibration data 1.) Analog Input(+/-5V) 2.) Analog Input(0~5V) 3.) Analog Output(0~5V) 4.) Read All DI channels 5.) Read One DI channel 6.) Write All DO channels 7.) Write One DO channel 8.) DO Readback All channels 9.) DO Readback One channel 10.) Exit Note1:AI(Ch0~5);AO(Ch0);DO(Ch0~3);DI(Ch0~3)	Please choose one of the following : 0.) Read the calibration data 1.) Analog Input(+/-5V) 2.) Analog Input(0~5V) 3.) Analog Output(0~5V) 4.) Read All DI channels 5.) Read One DI channel 6.) Write All DO channels 7.) Write One DO channel 8.) DO Readback All channels 9.) DO Readback One channel 10.) Exit Note1:AI(Ch0~5);AO(Ch0);DO(Ch0~3);DI(Ch0~3)
Choose an option and press [Enter]: 9	Choose an option and press [Enter]: 1
DO Channel (0~3): 0 DOO => ON	Channel (0~5): 0 A11 => 4.0175V

4.3 RS-232/422/485, DI/O Expansion Boards

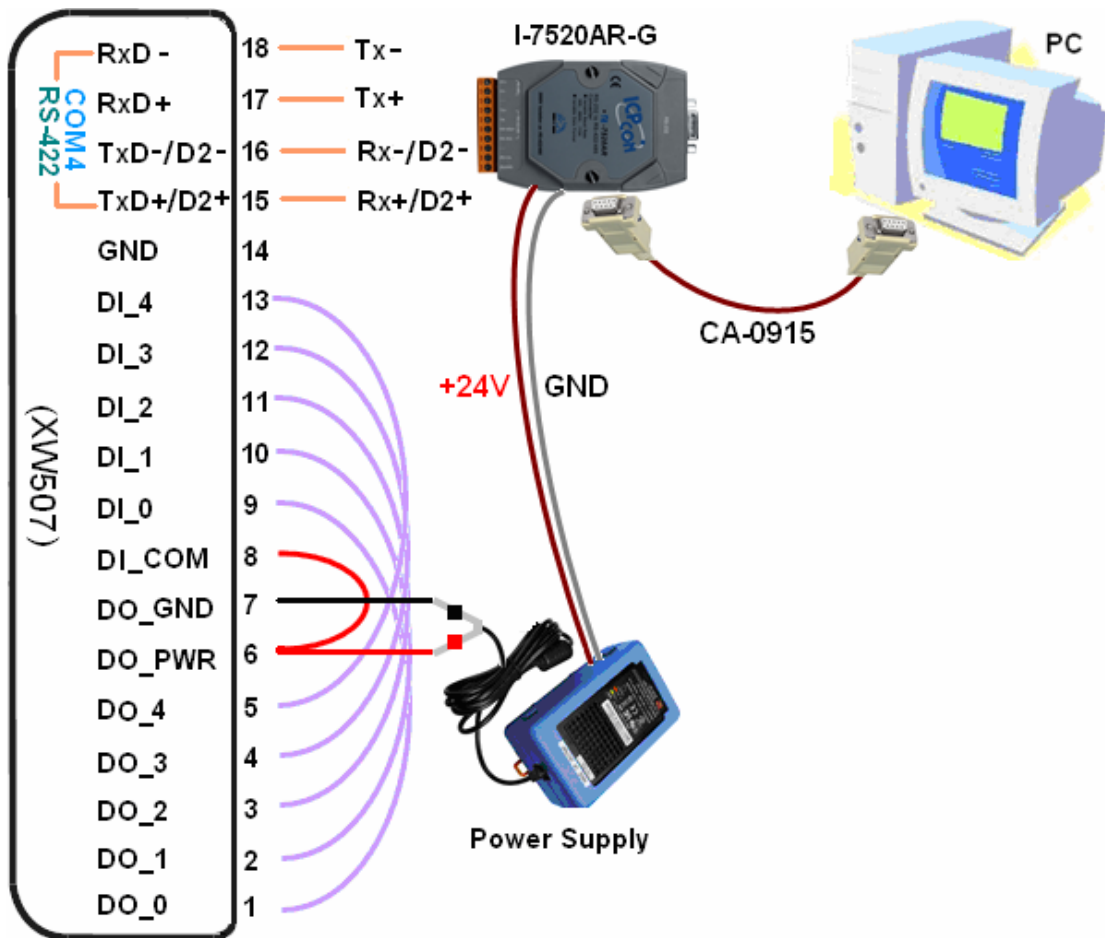
4.3.1 RS-422, DIO

~Available soon~

➔ Location

To install LinPAC-51XX SDK first from LinPAC-51XX's CD or FTP site of ICP DAS, and user can refer to C:\cygwin\LinCon8k\examples\xwboard\xw5xx\xw507.c

➔ Wire Connection



➔ Run

Part I

Step1: Download xw507.exe in LinPAC-51XX.

Step2: Changes the permission of a file as below:

```
# chmod 755 xw507.exe
```

Step3: Running program.

```
# ./xw507.exe
```

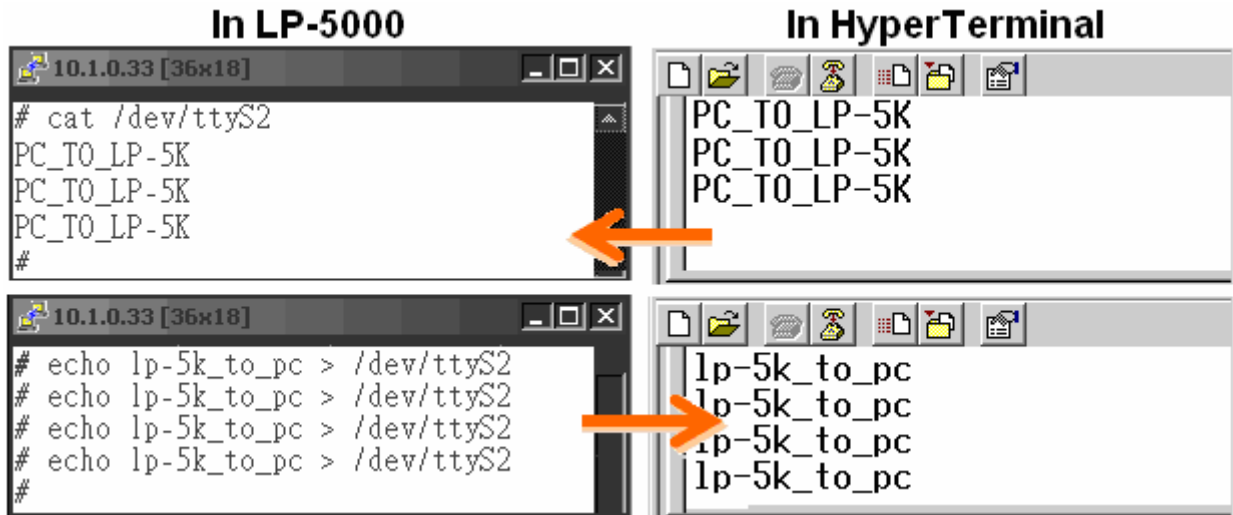
Part II

Step1: Telnet to LinPAC-51XX.

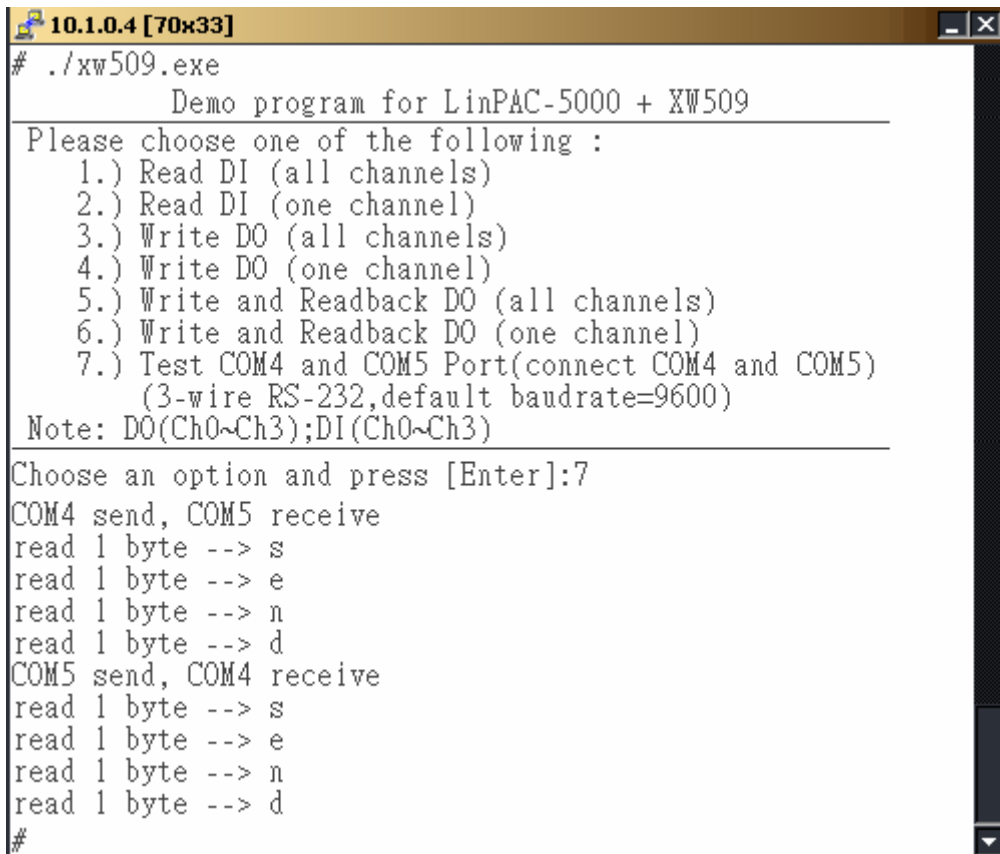
Step2: Start HyperTerminal by clicking on 'Start → Programs → Accessories → Communications → Hyper Terminal'

Step3: In the 'COM properties' dialog box, please set for 115200 bits per second, 8 data bits, no parity, 1 stop bit and no flow control to set up the communication parameters for the COM1 port, and press 'OK' when done.

Step4: Send message to the COM4 port from LP-5000 or HyperTerminal respectively.



Note: User can also refer to C:\cygwin\LinCon8k\examples\xwboard\xw5xx\xw509.c



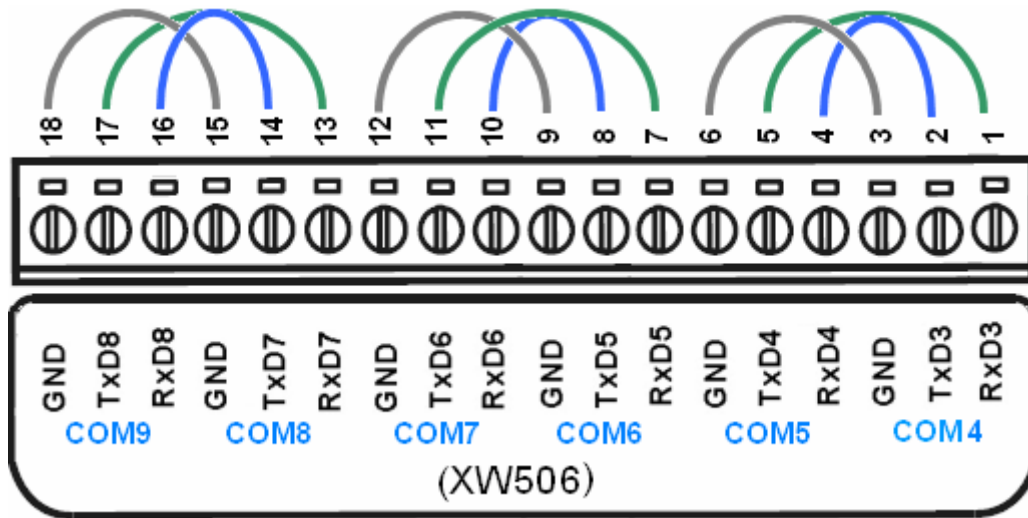
4.3.2 RS-232

➤ Location

To install LinPAC-51XX SDK first from LinPAC-51XX's CD or FTP site of ICP DAS, and user can refer to C:\cygwin\LinCon8k\examples\xwboard\xw5xx\sio_port.c

Part I

➤ Wire Connection



➤ Run

Step1: Download sio_port.exe in LinPAC-51XX.

Step2: Changes the permission of a file as below:

```
# chmod 755 sio_port.exe
```

Step3: Running program.

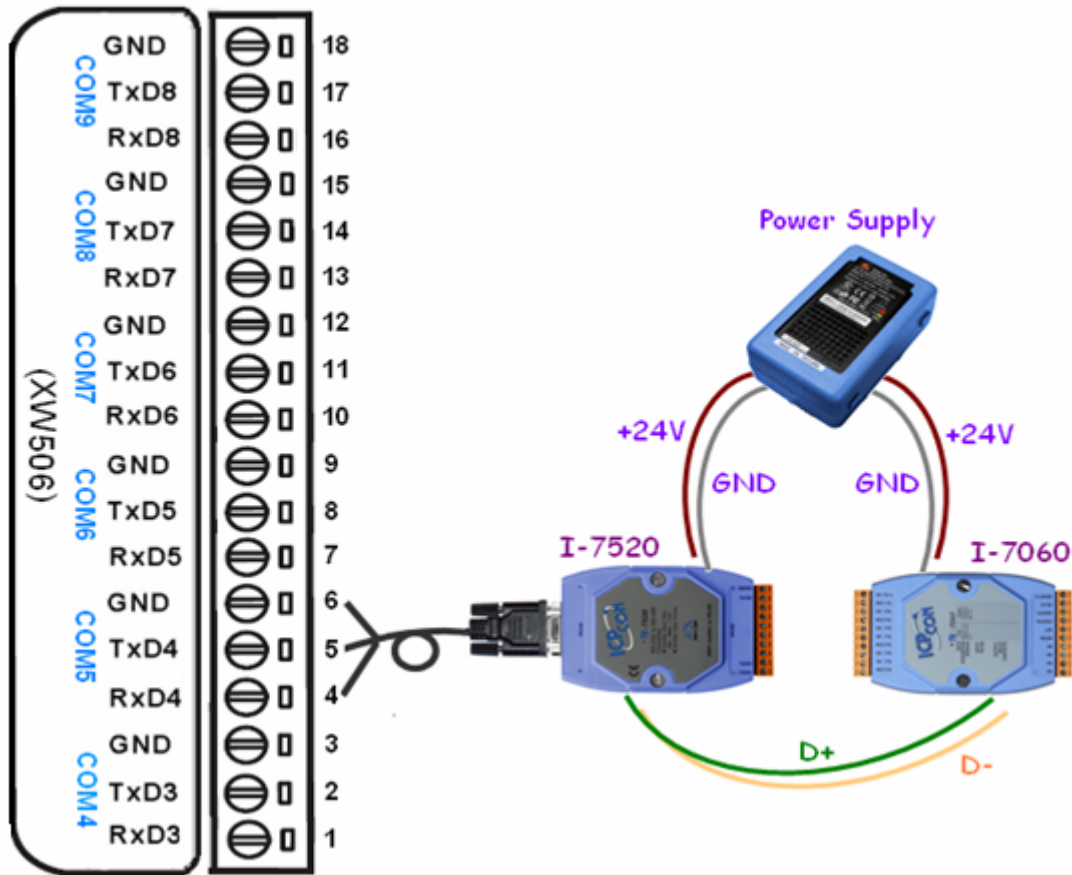
```
# ./sio_port.exe
```

Note: The demo is define: COM4 send, COM5 receive
COM6 send, COM7 receive
COM8 send, COM9 receive



Part II

➡ Wire Connection



➡ Run

Step1: Using 'setexdo' command to test COM5

Step2: Type **"setexdo 0 5 8 115200 1"** and make RL4 ON.

```

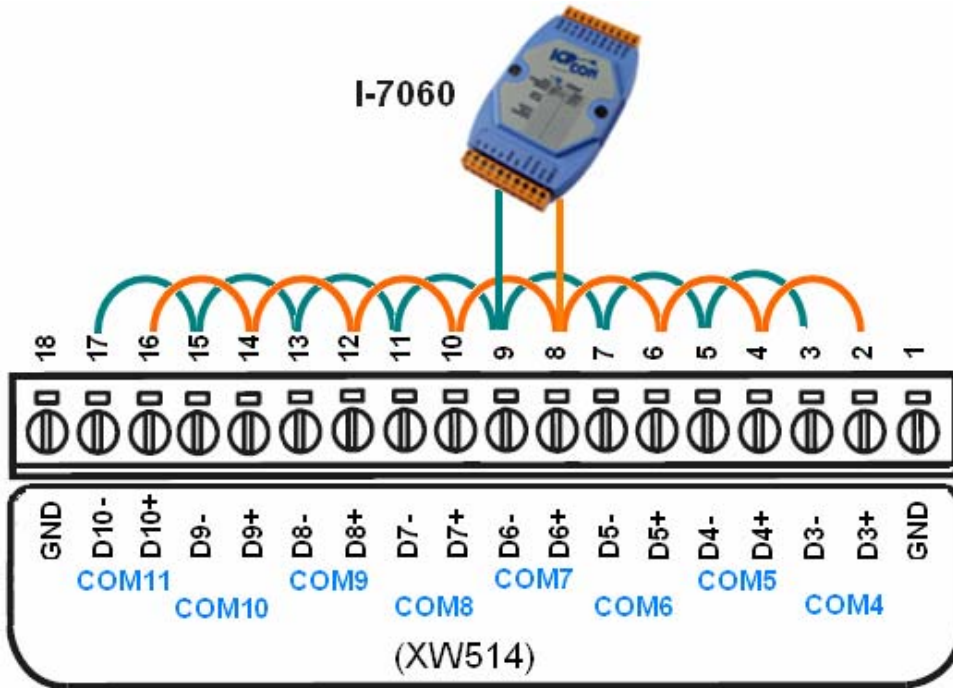
10.1.0.4 [70x23]
# setexdo
ICPDAS iTalk utility v15
function : setexdo
Set digital output value to a module
Usage: setexdo slot l data
      setexdo slot comport data baudrate address
Example 1:setexdo 2 1 55
Set the dec digital output value to the module at slot 2
Example 2:setexdo 0 3 55 9600 2
Set the dec digital output value to the module at COM3
#
#
# setexdo 0 5 8 115200 1 → Connect COM5 to I-7060 with I-7520,
#                               and Make RL4 ON
#
    
```

4.3.3 RS-485

➔ Location

To install LinPAC-51XX SDK first from LinPAC-51XX's CD or FTP site of ICP DAS, and user can refer to C:\cygwin\LinCon8k\examples\xwboard\xw5xx\

➔ Wire Connection



➔ Run

Step1: Using 'getsendreceive' command to test COM4

Step2: Type "getsendreceive 0 4 1 '\$01M' 9600" and receive response: !017060

```
10.1.0.4 [70x23]
# getsendreceive 0 4 1 '$01M' 9600
!017060#
# getsendreceive 0 5 1 '$01M' 9600
!017060#
# getsendreceive 0 6 1 '$01M' 9600
!017060#
# getsendreceive 0 7 1 '$01M' 9600
!017060#
# getsendreceive 0 8 1 '$01M' 9600
!017060#
# getsendreceive 0 9 1 '$01M' 9600
!017060#
# getsendreceive 0 10 1 '$01M' 9600
!017060#
# getsendreceive 0 11 1 '$01M' 9600
!017060#
```

🔍 Location

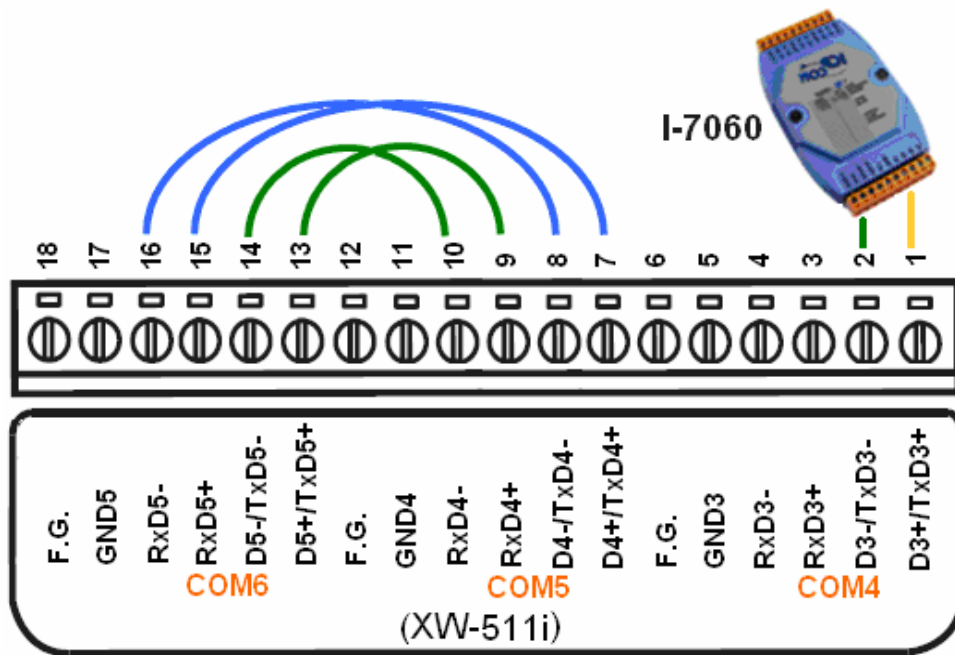
To install LinPAC-51XX SDK first from LinPAC-51XX's CD or FTP site of ICP DAS, and user can refer to C:\cygwin\LinCon8k\examples\xwboard\ xw5xx\

🔌 Wire Connection

Part1: COM4 connect with I-7060 (Check the I-7060 configuration value is 9600/8/n/1)

Part2: COM5 send message and COM6 receive

Part3: COM6 send message and COM5 receive



🏃 Run

Part I

There are more than six instructions in `/usr/local/bin` directory.

Step1: Using `'getsendreceive'` command to test COM4

Step2: Type `"getsendreceive 0 4 1 '$01M' 9600"` and receive response: `!017060`

```

10.1.0.4 [70x23]
# getsendreceive
ICPDAS iTalk utility v15
function : getsendreceive
Send ASCII command and wait resopnse from a serial module
Usage: getsendreceive slot 1 timeout command
      getsendreceive slot comport timeout command baudrate
Example 1:getsendreceive 2 1 1 '$00M'
Send command $00M to the module at slot 2 and wait response
Example 2:getsendreceive 0 3 1 '$01M' 9600
Send command $01M to the module at COM3 and wait response
#
# getsendreceive 0 4 1 '$01M' 115200
!017060#
    
```

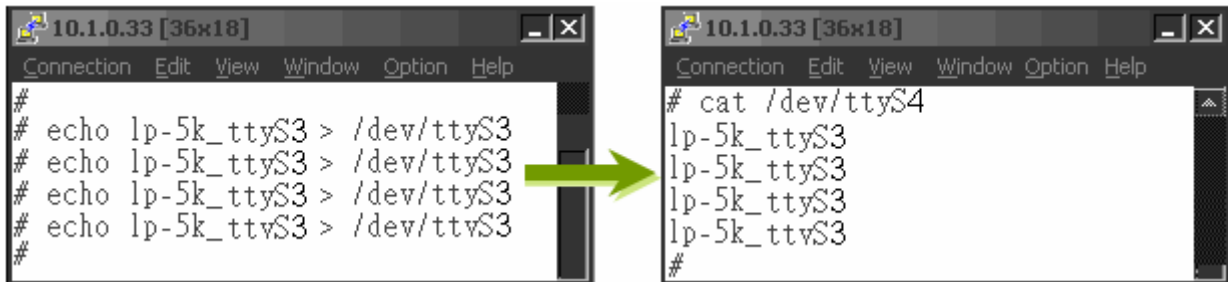
The following is the result of a 'setexdo' command of i-Talk utility .

```
10.1.0.4 [70x23]
installed XW-boards list
slot 1 ... XW-511
# setexdo
ICPDAS iTalk utility v15
function : setexdo
Set digital output value to a module
Usage: setexdo slot 1 data
      setexdo slot comport data baudrate address
Example 1:setexdo 2 1 55
Set the dec digital output value to the module at slot 2
Example 2:setexdo 0 3 55 9600 2
Set the dec digital output value to the module at COM3
# setexdo 0 4 2 115200 1
#
```

Part II

Step1: Open two Telnet connections (COM5 send, COM6 receive).

Step2: In one of connections, type 'cat /dev/ttyS4' for receive message from COM5 port, and another send message by 'echo' command to the COM5 port (/dev/ttyS3).



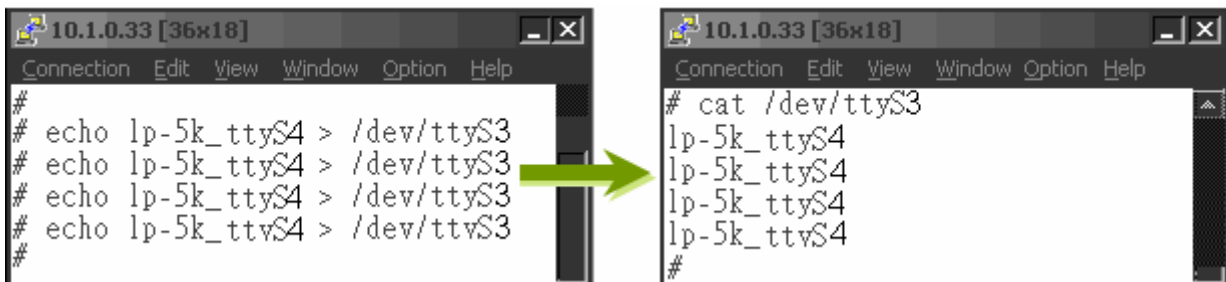
```
10.1.0.33 [36x18]
Connection Edit View Window Option Help
#
# echo lp-5k_ttyS3 > /dev/ttyS3
# echo lp-5k_ttyS3 > /dev/ttyS3
# echo lp-5k_ttyS3 > /dev/ttyS3
# echo lp-5k_ttyS3 > /dev/ttyS3
#

10.1.0.33 [36x18]
Connection Edit View Window Option Help
# cat /dev/ttyS4
lp-5k_ttyS3
lp-5k_ttyS3
lp-5k_ttyS3
lp-5k_ttyS3
#
```

Part III

Step1: Open two Telnet connections (COM6 send, COM5 receive).

Step2: In one of connections, type 'cat /dev/ttyS3' for receive message from COM6 port, and another send message by 'echo' command to the COM6 port (/dev/ttyS4).



```
10.1.0.33 [36x18]
Connection Edit View Window Option Help
#
# echo lp-5k_ttyS4 > /dev/ttyS3
# echo lp-5k_ttyS4 > /dev/ttyS3
# echo lp-5k_ttyS4 > /dev/ttyS3
# echo lp-5k_ttyS4 > /dev/ttyS3
#

10.1.0.33 [36x18]
Connection Edit View Window Option Help
# cat /dev/ttyS3
lp-5k_ttyS4
lp-5k_ttyS4
lp-5k_ttyS4
lp-5k_ttyS4
#
```