# *Modbus RTU／TCP Server* User Manual

## LinPAC／LinCon

*Version 1.0*

This document applied to models :

LinPAC-8000(Linux kernel 2.6.19 based)

LinPAC-8x81(Linux kernel 2.6.18 based)

LinCon-8000(Linux kernel 2.4.21 based)

.

# CONTENTS

# CH 1 Introduction

This manual is written for modbus users of *ICPDAS LinPAC and LinCon series*, and it only supports 8K & 87K I/O Modules.

This document will guid you :

- How to setup Modbus Server on your devices.
- How to consist your modbus requests.
- How to interpret response messages.

# 1-1. Install Modbus server on your LinPAC／LinCon

## ▶ Software Installation

If your LinPAC／LinCon is former released, you have to install the Modebus Server by manual.

**STEP 1 :**

Download the latest Modbus Server for LinPAC／LinCon.

http://www.icpdas.com/download/download-list.htm

**STEP 2 :**

Copy the "mbserver10" file to your LinPAC／LinCon, and change the file permission to "755" or above.

*#chmod 755 ./mbserver10*

**STEP 3 :**

Before executing LinPAC／LinCon Modbus Server, you have to check what's the index number of your TTY devices.

*#dmesg | grep 'ttyS'*

In general, LinCon support 2 serial ports,

ttyS0 : COM2 RS-232 (Recommend!)

ttyS1 : COM3 RS-485

and 3 serial ports are availabled on LinPAC.

ttyS0 : COM1, RS232 (Reserved for console terminal)

ttyS1 : COM3, RS232／RS-485

ttyS34 : COM4, RS232 (Recommend!)

**STEP 4 :**

You can use following command to start Modbus Server.

*#./mbserver10 [Device Net ID] [COM port Index number]*

For example :

#./mbserver10 4 0 → Start Modbus Server for ttyS0, and the Device ID = 4

#./mbserver10 9 34 → Start Modbus Server for ttyS34, and the Device ID = 9

To check the command usage :

*#./mbserver10*

***NOTE :***

The LinPAC／LinCon Device Net ID must be the unique of the network.

For RTU protocol, the default COM port control setting is:
Baud rate : **115200** ,
Parity : **None**
Data bits : **8**
Stop bits : **1**

For TCP protocol, you can use "ifconfig" command to get the IP address of Modbus Sever.

.

**Running automatically at boot time :**

**STEP 1** : Copy "mbserver" file to /etc/init.d of your LinPAC／LinCon

**STEP 2** : Edit "mbserver" script file,
specify "mbserver10" directory and modify parameters to fit your system.

**STEP 3** : Creat a symbolic link in /etc/rc2.d directory
ln –s /etc/init.d/mbserver /etc/rc2.d/S99MBServer

***NOTE :***

*Modbus server must be executed after serial port initialization.*

## ▶ Hardware Connection

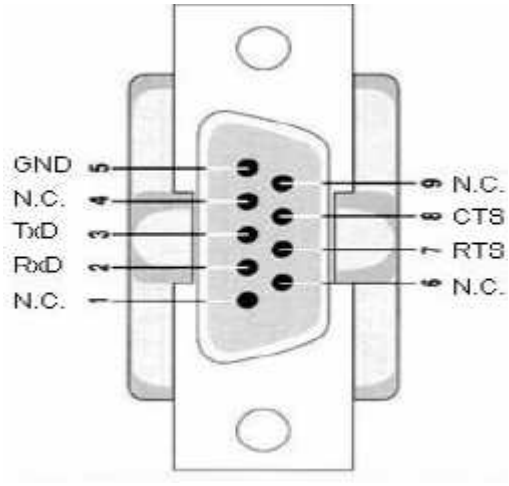
LinPAC／LinCon Modbus Server supports Modbus RTU and Modbus TCP protocol.

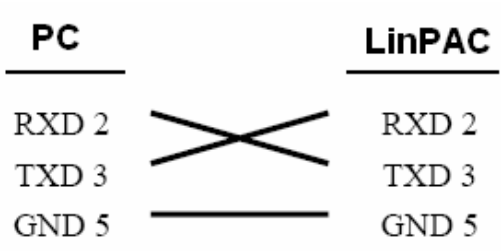
**For TCP protocol**

You can connect modbus client and server by using the ethernet cable with RJ-45 connector.

**For RTU protocol**

9 pins D-sub cable is necessary. The pin assignment is :




*Ordering Information :*

CA-0910F：9-pin Female-Female D-sub cable, 1M Cable

# CH 2 Modbus Control Function

This chapter is about how to consist your modbus requests, and how to interpret response messages.

LinPAC／LinCon Modbus Server supports functions:

| Function Code | Sub Code | Description | Section |
|---|---|---|---|
| 01 | - | Read back multiple coils status (0xxxx) for DO | 2-1 |
| 02 | - | Read multiple input discretes (1xxxx) for DI | 2-2 |
| 03 | - | Read back multiple registers (4xxxx) for AO | 2-3 |
| 04 | - | Read multiple input registers (3xxxx) for AI | 2-4 |
| 05 | - | Write single coil (0xxxx) for DO | 2-5 |
| 06 | - | Write single register (4xxxx) for AO | 2-6 |
| 15 | - | Force multiple coils (0xxxx) for DO | 2-7 |
| 16 | - | Write multiple registers (4xxxx) for AO | 2-8 |
| 108 | 7 | Read range code of ICPDAS AI/O module | 2-9.1 |
| | 8 | Set range code of ICPDAS AI/O module | 2-9.2 |

In following sections, the defined **Protocol Data Unit (PDU)** is used within the framework of Modbus RTU. For Modbus TCP protocol, there are 6 bytes of prefixed fields need be added.

| Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 ～ |
|---|---|---|---|---|---|---|
| Transaction identifier | | Protocol identifier | | PDU length | | PDU |
| Hi (0) | Lo (0) | Hi (0) | Lo (0) | Hi | Lo | Formate is defined in eache section |

***NOTE :***

There are some address rules need to be noted :

(1.) Slot index start from 1, and max. is 8. (depended on your LinPAC／LinCon modules )

(2.) DI/O and AI/O points are dressed starting from 0.

## 2-1. Function Code 01：Read Coils (0xxxx)

This function code is used to read back the ON/OFF settings of DOs.

### ▶ Request PDU

| Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
|---|---|---|---|---|---|
| Device Net ID | Function code | Address Hi | Address Lo | Data Count Hi | Data Count Lo |
| ID [*] | 01 | Slot Index (1~8) | DO Address | Contiguous DO bits for query | |

[*] : Device Net ID is specified when you start LinPAC／LinCon Modbus Server, and it must be the unique of the network.

### ▶ Response PDU

| Byte 0 | Byte 1 | Byte 2 | Byte 3 ～ (Byte Count + 2) |
|---|---|---|---|
| Device Net ID | Function code | Byte count = (bit count + 7) / 8 | Response coils data |
| ID | 01 | Response data length | DO ON / OFF status [**] |

[**] Least Significant Bit (LSB) is the first coil.
If the data bits is not a multiple of eight, the remaining bits will be padded with zeros.

Here is an example of a query to read DO #0 ~ #12 in slot 4 of LinPAC (Net ID 9) :

| RTU Request | | | Response | |
|---|---|---|---|---|
| Device Net ID | | 09 | Device Net ID | 09 |
| Function Code | | 01 | Function Code | 01 |
| Address Hi (Slot index 1~8) | | 04 | Data length (Byte) | 02 |
| Address Lo (DO start address) | | 00 | Coils Data (DO address 8 ~ 12) | 01 |
| Data Count | Hi | 00 | Coils Data (DO address 0 ~ 7 ) | E3 |
| | Lo | 0C | | |

## 2-2. Function Code 02： Read Discrete Inputs (1xxxx)

This function code is used to read the ON/OFF status of DIs.

**▶ Request PDU**

| Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
|---|---|---|---|---|---|
| Device Net ID | Function code | Address Hi | Address Lo | Data Count Hi | Data Count Lo |
| ID | 02 | Slot Index (1~8) | DI Address | Contiguous DI bits for query | |

**▶ Response PDU**

| Byte 0 | Byte 1 | Byte 2 | Byte 3 ～ (Byte Count + 2) |
|---|---|---|---|
| Device Net ID | Function code | Byte count = (bit count + 7) / 8 | Response DI value |
| ID | 02 | Response data length | DI ON / OFF status [**] |

[**] Least Significant Bit (LSB) is the first DI value.

If the data bits is not a multiple of eight, the remaining bits will be padded with zeros.

## 2-3. Function Code 03：Read Holding Registers (4xxxx)

This function code is used to read back AO settings from I-8017 and I-87017 modules.

**▶ Request PDU**

| Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
|---|---|---|---|---|---|
| Device Net ID | Function code | Address Hi | Address Lo | Data count Hi | Data count Lo |
| ID | 03 | Slot Index (1~8) | AO Address | Contiguous AO no. for query | |

***I-8024 Module NOTE ：***

The bytes of AO address are invalid for module I-8024, no matter which AO address you specified, it only allows to read back the latest output value for response.

▶ **Response PDU**

| Byte 0 | Byte 1 | Byte 2 | Byte 3 ～ (Byte Count + 2) |
|---|---|---|---|
| Device Net ID | Function code | Byte count | Response AO values |
| ID | 03 | Response data length [**] | two bytes / pre AO register |

[**] AO value is presented in a WORD signature = 2 Bytes.

***NOTE*** **：**

Real value on the I/O module = (Response AO value) / 1000

Minus value (>0x7FFF) is presented in 2's Complement.

Here is an example of a query to read AO #3 ~ #4 in slot 5 of LinPAC (Net ID 6) :

| Request | | | Response | | |
|---|---|---|---|---|---|
| Device Net ID | | 06 | Device Net ID | | 06 |
| Function Code | | 03 | Function Code | | 03 |
| Address Hi (Slot index 1~8) | | 05 | Data length (Byte) | | 04 |
| Address Lo (AO start address) | | 03 | AO #3 read back value | Hi | 13 |
| Data Count | Hi | 00 | | Lo | 8F |
| | Lo | 02 | AO #4 read bacek value | Hi | EC |
| | | | | Lo | 78 |

From above example, the response value of AO#3 is 0x138F , then we can get the real value on the I/O module should be : 5007(=0x138F) / 1000 = 5.007 (V).

The response value of AO #4 is 0xEC79, and it's greater than 0x7FFF, that means we have a minus value. The 2's complement of 0xEC78 is 0x1388 (= 5000 d),
So we can get the real value on the I/O module : -5000 / 1000 = -5.000 (V)

## 2-4. Function Code 04：Read Input Registers (3xxxx)

This function code is used to read AI values from I-8017 and I-87017 modules.

▶ **Request PDU**

| Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
|---|---|---|---|---|---|
| Device Net ID | Function code | Address Hi | Address Lo | Data count Hi | Data count Lo |
| ID | 04 | Slot Index (1~8) | AI Address | Data type[*] | AI count |

[*] I-8017／I-87017 supports 5 range modes for option, you can specify following type code to interpret your response data.

0: depend on the current setting of I/O module.

1: +/- 10.0V,

2: +/- 5.0V,

3: +/- 2.5V,

4: +/- 1.25V,

5: +/- 20mA.

You can read current range setting by using function code 108 and sub function code 7. For more detail, please refer to the Function Code 108 session.

▶ **Response PDU**

| Byte 0 | Byte 1 | Byte 2 | Byte 3 ～ (Byte Count + 2) |
|---|---|---|---|
| Device Net ID | Function code | Byte count | Response AI values |
| ID | 04 | Response data length [**] | two bytes / pre AI register |

[**] AI value is presented in a WORD signature = 2 Bytes.

## 2-5. Function Code 05：Write Single Coil (0xxxx)

This function code is used to write the ON/OFF status for single DO point.

▶ **Request PDU**

| Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
|---|---|---|---|---|---|
| Device Net ID | Function code | Address Hi | Address Lo | Set value Hi | Set value Lo |
| ID | 05 | Slot Index (1~8) | DO Address | FF／00 [*] | 00 |

[*] 0xFF to trun ON the coil, 0x00 to trun OFF the coil.

▶ **Response PDU**

| Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
|---|---|---|---|---|---|
| Device Net ID | Function code | Address Hi | Address Lo | eched value Hi | eched value Lo |
| ID | 05 | Slot Index (1~8) | DO Address | FF／00 | 00 |

## 2-6. Function Code 06：Write Single Registers (4xxxx)

This function code is used to write single AO for I-8024 and I-87024 modules.

**▶ Request PDU**

| Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
|---|---|---|---|---|---|
| Device Net ID | Function code | Address Hi | Address Lo | Data value Hi | Data value Lo |
| ID | 06 | Slot Index (1~8) | AO Address | Single register value | |

**▶ Response PDU**

| Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
|---|---|---|---|---|---|
| Device Net ID | Function code | Address Hi | Address Lo | Data value Hi | Data value Lo |
| ID | 06 | Slot Index (1~8) | AO Address | Echoed setting value | |

## 2-7. Function Code 15：Write Multiple Coils (0xxxx)

This function code is used to set ON/OFF status for sequence DOs.

**▶ Request PDU**

| Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7～ |
|---|---|---|---|---|---|---|---|
| Devic ID | FC | Addr. Hi | Addr.Lo | DO Bit Count | | Byte Count | Data value |
| ID | 0x0F | Slot no. (1~8) | DO Address | Hi | Lo | Data length | [*] |

[*] Least Significant Bit (LSB) is the first DO value.

**▶ Response PDU**

| Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
|---|---|---|---|---|---|
| Device Net ID | Function code | Address Hi | Address Lo | DO Bit Count | |
| ID | 0x0F | Slot Index (1~8) | AO Address | Hi | Lo |

## 2-8. Function Code 16：Write Multiple Registers (4xxxx)

This function code is used to set a sequence of AOs for I-8024 and I-87024 modules.

**▶ Request PDU**

| Byte 0 | Byte1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7~ |
|---|---|---|---|---|---|---|---|
| Devic ID | FC | Address Hi | Address Lo | AO count Hi | AO count Lo | Data Count | AO value |
| ID | 0x10 | Slot Index | AI Address | Contiguous AOs for query | | [*] | [**] |

[*] Data Count = AO Count * 2

[**] AO value is presented in a WORD signature = 2 Bytes.

**▶ Response PDU**

| Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
|---|---|---|---|---|---|
| Device Net ID | Function code | Address Hi | Address Lo | Contiguous AO Count | |
| ID | 0x10 | Slot Index (1~8) | AO Address | Hi | Lo |

## 2-9. Function Code 108：Device Configuration

This function code is definded by ICPDAS.

### 2-9.1 Sub Function Code 07 : Read range code of AI/O module

**▶ Request PDU**

| Byte 0 | Byte1 | Byte 2 | Byte 3 | Byte 4 |
|---|---|---|---|---|
| Devic Net ID | Function Code | Sub Function Code | Address Hi | Address Lo |
| ID | 0x6C | 0x07 | Slot Index (1~8) | I/O Address |

**▶ Response PDU**

| Byte 0 | Byte1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
|---|---|---|---|---|---|
| Devic ID | Func. Code | Sub Func. Code | Address Hi | Address Lo | Range code |
| ID | 0x6C | 0x07 | Slot Index | I/O Address | [*] |

[*] Analog I/O modules defined various range code (type code) for different purposes. More detail definition, please refer to following website：

http://ftp.icpdas.com/pub/cd/8000cd/napdos/dcon/io_module/dcon/8k87k/modules/typetable.htm

### 2-9.2 Sub Function Code 07 : Set range code of AI/O module

#### ▶ Request PDU

| Byte 0 | Byte1 | Byte 2 | Byte 2 | Byte 3 | Byte |
|---|---|---|---|---|---|
| Devic ID | Func. Code | Sub Func. Code | Address Hi | Address Lo | Set value |
| ID | 0x6C | 0x08 | Slot Index (1~8) | I/O Address | Range code |

#### ▶ Response PDU

| Byte 0 | Byte1 | Byte 2 | Byte 2 | Byte 3 | Byte |
|---|---|---|---|---|---|
| Devic ID | Func. Code | Sub Func. Code | Address Hi | Address Lo | Echoed value |
| ID | 0x6C | 0x08 | Slot Index (1~8) | I/O Address | Range code |