

如何在 LinPAC 上使用 Modbus 協定存取 M-7000 遠端 I/O 模組?

適用於：			No. L2-002
平台	作業系統	作業系統版本	類別
所有 LinPACs	所有版本	所有版本	安裝和配置

Modbus 是 Modicon 公司於 1979 年為其產品所發表的通訊協定，目前已成為工業領域常採用的標準且廣泛用於工業電子裝置之間的連接。Modbus 通訊協定的維護及更新目前由 Modbus Organization 負責，更詳細的資訊請參考 <https://www.modbus.org/>。

M-7000 系列模組採用 Modbus RTU 協定，並使用 RS-485 介面來進行連接，適用的鮑率範圍為 1200 bps 到 115200 bps。使用的資料格式固定為：無同位元檢查(no parity)，8 個資料位元(data bits)，1 個停止位元(stop bit)。

本文敘述如何透過 LinPAC 控制 M-7000 系列模組，將分別就 LinPAC SDK 及 libmodbus 兩者提供的函數進行介紹，用於範例的模組為 M-7002 及 M-7060，使用 COM2 (/dev/ttyS0)接至 LinPAC。

注意：

DCON Utility 也可支援 M-7000 系列模組，請在選擇連接設定時勾選“Modbus RTU”方框來搜尋模組，用戶可於下方連結找到更多資訊：

http://www.icpdas.com/en/product/guide+Software+Utility_Driver+DCON_Utility_Pro



I. LinPAC SDK

用戶可以利用 LinPAC 內建程式來使用 M-7000 模組，LinPAC SDK 也提供了該程式範例的原始檔。表 1 列出了 Modbus 的基本功能碼及其對應敘述：

功能碼 (Function Code)	敘述
01 (0x01)	讀取一組數位輸出的值
02 (0x02)	讀取一組數位輸入的值
03 (0x03)	讀取一組類比輸出的值
04 (0x04)	讀取一組類比輸入的值
05 (0x05)	寫入一個數位輸出的值
06 (0x06)	寫入一個類比輸出的值
15 (0x0F)	寫入多個數位輸出的值
16 (0x10)	寫入多個類比輸出的值

表 1

用戶可以從 M-7000 系列模組的使用手冊查詢其適用的功能碼。

使用 **getmodbus** 指令傳送功能碼 01 到 04 的訊息至模組，如圖 1。

範例一：經由 COM2 埠，讀取 M-7002 的 AIO 值，使用功能碼 04，其位址為 01，鮑率為 115200 bps。

指令： # **getmodbus** **comport** **baudrate** **netid** **command** **addr** **count** **timeout**

```

root@icpdas: ~
root@icpdas:~# ./getmodbus 2 115200 1 4 0 1 100
16491root@icpdas:~#

```

圖 1

使用 **setmodbus** 指令傳送功能碼 05、06、15 以及 16 的訊息至模組，如圖 2。

範例二：經由 COM2 埠，對 M-7060 的 DO0~DO3 值寫入 10 (1010₂)，使用功能碼 15，其位址為 02，鮑率為 115200 bps。

指令： # **setmodbus** **comport** **baudrate** **netid** **command** **addr** **count** **value** **timeout**

```

root@icpdas: ~
root@icpdas:~# ./setmodbus 2 115200 2 15 0 4 10 100
root@icpdas:~#

```

圖 2

上方 2 個程式皆使用函數 **modbusRequest** 傳送 Modbus RTU 訊息至模組：

```
int modbusRequest (char cPort, char cNetID, char cFunction, WORD wAddr,
                  WORD wCount, char szBuf, WORD wBufLen, WORD wTimeout, WORD *wT)
```

參數：

cPort: [Input] 串列埠號碼
 cNetID: [Input] 設備的位址，M-7000 系列模組的預設位址為'1'
 cFunction: [Input] Modbus RTU 通訊協定的功能碼
 wAddr: [Input] 資料起始位址，使用十進位格式
 wCount: [Input] 資料 Byte 數
 szBuf[]: [Input/Output] 用於紀錄傳送/接收訊息時的資料
 wBufLen: [Input] szBuf[]的長度
 wTimeout: [Input] 逾時設定，一般設為 100ms
 *wT: [Output] 傳送訊息至收到回應所經過的時間，單位為 ms

返回值：

0: 函數成功執行
 其他: 執行失敗

使用功能碼 **70(0x46) Read/Write Module Settings** 來讀取/修改 M-7000 系列模組的設定，表 2 列出了 M-7000 系列模組共通的子功能碼，用戶可以針對特定模組查詢其使用手冊來了解更多支援的子功能碼：

子功能碼 (Sub Function Code)	敘述
00 (0x00)	讀取模組名稱
04 (0x04)	設定模組位址
05 (0x05)	讀取模組連接設定
06 (0x06)	更改模組連接設定
32 (0x20)	取得目前軟體版本

表 2

注意：

在使用功能碼 **70(0x46) Read/Write Module Settings** 呼叫函數 **modbusRequest** 時，請使用子功能碼的值來替換 wAddr 的值，並且將 wCount 的值設為 0：

```
int modbusRequest (char cPort, char cNetID, 70, WORD wSubFunction, 0,
                  char szBuf, WORD wBufLen, WORD wTimeout, WORD *wT)
```

II. libmodbus

libmodbus 是一款免費的套件，可以用於設備間傳送/接收 Modbus 訊息。此套件以 C 語言寫成，並且可以在 Linux、Mac OS X、FreeBSD、QNX 以及 Windows 使用。詳細資訊請參考官網：www.libmodbus.org

簡易範例

```

modbus_t *ctx;
int rc;
uint8_t *tab_bits;

// COM2 = /dev/ttyS0, Baud rate = 115200, N = no parity, data bits = 8 bits, stop bits = 1 bit
ctx = modbus_new_rtu("/dev/ttyS0", 115200, 'N', 8, 1);
modbus_set_slave(ctx, SERVER_ID); // 設定欲連接模組的位址
if (modbus_connect(ctx) == -1) {
    fprintf(stderr, "Connection failed: %s\n",
            modbus_free(ctx);
    return -1;
}

// 在此傳送訊息，所支援的函數列於表 3
rc = modbus_write_bit(ctx, 0, 1); // 函數 FC: 05
if (rc != 1) {
    printf("ERROR modbus_write_bit (%d)\n", rc);
}else{
    rc = modbus_read_bits(ctx, 0, 1, tab_bits); // 函數 FC: 01
    if (rc != 1)
        printf("ERROR modbus_write_bit (%d)\n", rc);
}
modbus_close(ctx);
modbus_free(ctx);

```

功能碼 (Function Code)	對應的函數名稱
01 (0x01)	int modbus_read_bits (modbus_t *ctx, int addr, int nb, uint8_t *dest)
02 (0x02)	int modbus_read_input_bits (modbus_t *ctx, int addr, int nb, uint8_t *dest)
03 (0x03)	int modbus_read_registers (modbus_t *ctx, int addr, int nb, uint16_t *dest)
04 (0x04)	int modbus_read_input_registers (modbus_t *ctx, int addr, int nb, uint16_t *dest)
05 (0x05)	int modbus_write_bit (modbus_t *ctx, int addr, int status)
06 (0x06)	int modbus_write_register (modbus_t *ctx, int addr, const uint16_t value)
15 (0x0F)	int modbus_write_bits (modbus_t *ctx, int addr, int nb, const uint8_t *src)
16 (0x10)	int modbus_write_registers (modbus_t *ctx, int addr, int nb, const uint16_t *src)
17 (0x11)	int modbus_report_slave_id (modbus_t *ctx, int max_dest, uint8_t *dest)

表 3

libmodbus 套件中，於 **tests** 資料夾內提供了一些用於測試的範例程式，用戶也可以自行修改以符合使用需求。圖 3 為 **tests** 資料夾裡 **random-test-client.c** 的執行結果，此程式碼有經過一些修改以印出更多訊息，所使用的模組為 M-7060，有 4DI 及 4DO。

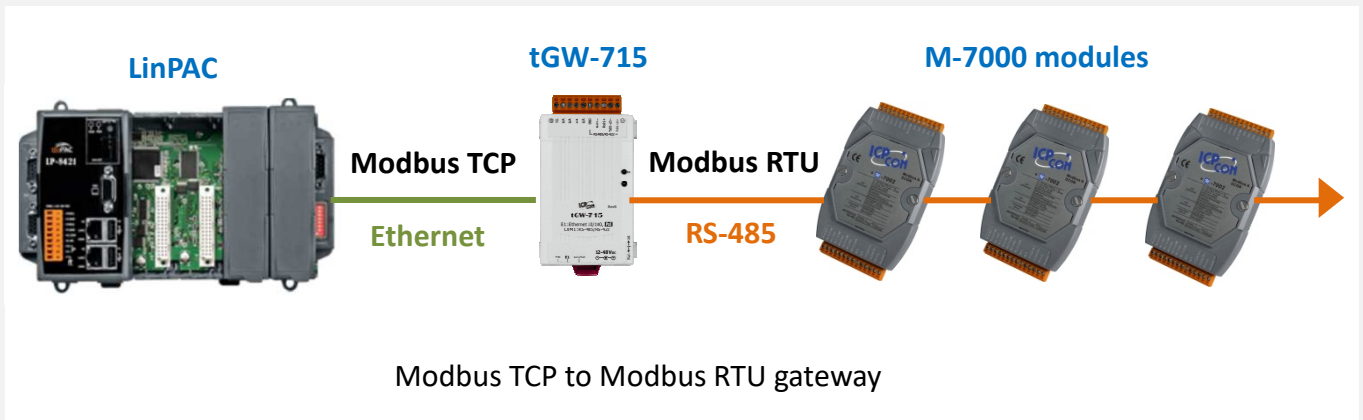
```

root@icpdas: ~
root@icpdas:~# ./random-test-client
<DO test>
modbus_write_bit test:
success
modbus_read_bits single test:
success
modbus_write_bits test:
success
modbus_read_bits test:
success

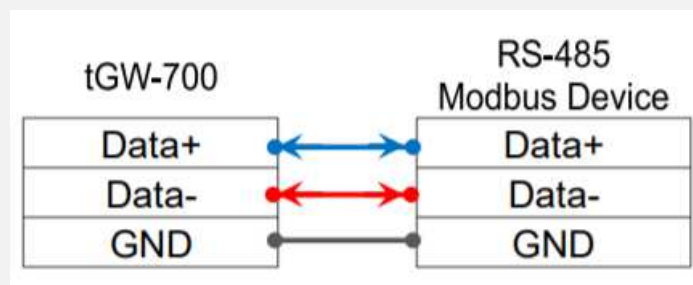
```

圖 3

注意：若用戶較偏好使用 Modbus TCP，請在 LinPAC 及 M-7000 系列模組之間使用 **Modbus TCP to RTU gateway** 相連。



Modbus TCP to RTU gateway 與 M-7000 系列模組的接線方式如下：



使用 **modbus_t* modbus_new_tcp(const char *ip, int port)** 來代替函數 **Modbus_new_rtu()**，其中 port 值應設為 502。