

Modbus Development kit for Linux



V1.0.1 June 2024
Edited by Cindy Huang

保固說明

泓格科技股份有限公司 (ICP DAS) 所生產的產品，均保證原始購買者對於有瑕疵之材料，於交貨日起保有為期一年的保固。

免責聲明

泓格科技股份有限公司對於因為應用本產品所造成的損害並不負任何法律上的責任。本公司保留有任何時間未經通知即可變更與修改本文件內容之權利。本文所含資訊如有變更，恕不予另行通知。本公司盡可能地提供正確與可靠的資訊，但不保證此資訊的使用或其他團體在違反專利或權利下使用。此處包涵的技術或編輯錯誤、遺漏，概不負其法律責任。

版權所有

2023 © 泓格科技股份有限公司版權所有。

商標識別

本文件提到的所有公司商標、商標名稱及產品名稱分別屬於該商標或名稱的擁有者所有。

技術服務

如有任何問題，請與本公司客服聯絡，我們將盡速為您服務。

E-mail 信箱：service@icpdas.com

Contents

1. 開始入門	4
1.1. Linux SDK 簡介	4
1.2. 下載 LinPAC SDK	5
2. Linux Modbus 開發工具	6
2.1 LinPAC 與 C 應用	6
2.1 LinPAC 與 Python 應用	9
2.2 LinPAC 與 Perl 應用	11
附錄 A. 如何在 Windows 平台上編譯包含 libmodbus 函式庫的應用程式	15
附錄 B. 修訂記錄	23

1. 開始入門

本章節將介紹首次使用 LinPAC SDK 的用戶了解、下載、安裝和配置基本程序所需的步驟。

1.1. Linux SDK 簡介

本節將討論 LinPAC SDK 中採用的一些技術，包括描述如何輕鬆使用 LinPAC SDK 的詳細說明。

以 AM335x SDK 為例，LinPAC SDK 基於 Cygwin，也是 Microsoft Windows 系統的類 Linux 環境，不但提供了強大的 GCC 交叉編譯器和 IDE (集成開發環境)，可以快速開發 LinPAC 應用。因此，一旦創建了應用程序，便可使用 LinPAC SDK，將其編譯成可在 LinPAC 嵌入式控制器上運行的可執行文件。

目前 LinPAC SDK 的編譯操作方式，依產品分做二大類：

類別 步驟	LP-8x41/51xx LP-2241M/5231/8x2x/9x2x	LP-2841M/LX-8000/LX-9000
0.	下載 SDK，安裝於 Windows 或 Linux 個人電腦	下載 SDK，安裝於 LinPAC
1.	於 SDK 目錄內找到 'helloworld.c' 範例	於 SDK 目錄內找到 'helloworld.c' 範例
2.	利用 SDK 提供的編譯工具，於 Windows 或 Linux 個人電腦，編譯範例	直接在 LinPAC 裡編譯範例
3.	上傳範例至 LinPAC，並執行	直接執行範例

1.2. 下載 LinPAC SDK

LinPAC 系列產品是以 Linux 開發的控制器，支持 DCON 和 Modbus 協議，用戶可以從泓格科技的產品網站安裝 LinPAC SDK。下表列出了 Modbus 工具的文件路徑，更多關於 LinPAC SDK 的信息，請參閱網站產品資訊。

LinPAC		下載路徑
PXA270	LP-8x41	https://www.icpdas.com/en/download/show.php?num=982&model=LP-8441-EN
iMX8MM	LP-2841	https://www.icpdas.com/en/download/show.php?num=8723&model=LP-2841M
AM335x Series	LP-224x LP-52xx	https://www.icpdas.com/en/download/show.php?num=1195&model=LP-5231M
	LP-8x2x LP-9x2x	https://www.icpdas.com/en/download/show.php?num=915&model=LP-9821
	X86/E38xx Series	LX-Series

▲注意

- 1) 上面有四種獨立的 LinPAC SDK，不同的 LinPAC 不能同時共享源文件、庫文件和編譯文件，用戶需要下載各自最新版本的 LinPAC SDK。
- 2) 我們建議用戶通過 'sudo' 或 'su' 命令將用戶 ID 更改為 **root**。
- 3) Linux 64 位元的操作系統，缺少向下相容 32 位元 Library 的問題：若您的 Linux 個人電腦是 64 位元，則在運行 32 位版本的 LinPAC SDK（Linux 版本）前，您必須先在系統上，自行下載、安裝 32 位元的支援套件。

以 AM335x SDK 為例，可至以下路徑位置找到 modbus 相關檔案：

作業系統平台	檔案路徑位置
解壓縮/安裝 .exe 檔至 C 槽	
Windows	C:\cygwin\LinPAC_AM335x_SDK\examples\xvboard\ 或 C:\cygwin\LinPAC_AM335x_SDK\examples\modbus\
解壓縮/安裝.bz2 檔至 root (/) 目錄	
Linux	root@LinuxPC-ICPDAS: /icpdas/linpac_am335x_sdk/i8k/examples/xvboard/ 或 root@LinuxPC-ICPDAS: /icpdas/linpac_am335x_sdk/i8k/examples/modbus/

有關 modbusRequest() API 函數的更多信息，請參考以下手冊：

https://www.icpdas.com/web/product/download/pac/linux/lp-5000/document/manual/xv-board_linux_api_reference_manual_en.pdf

2. Linux Modbus 開發工具

泓格科技 LinPAC 系列產品採用具有高彈性、開放性特色的 Linux 作業系統，您可以使用 DCON 或 Modbus 通訊協定來控制 tM 系列模組。本章節將以 Modbus 通訊協定為例，介紹三種軟體工具應用，提供您多元化的開發。

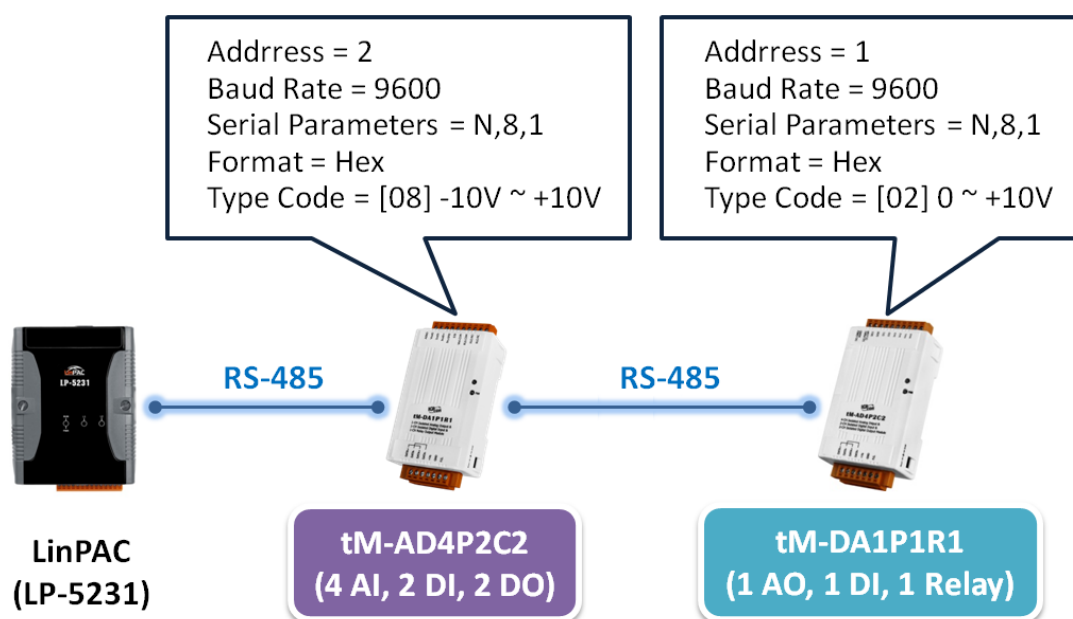
2.1 LinPAC 與 C 應用

LinPAC 系列產品支援 Modbus 與 DCON 通訊協定，用戶可至 ICP DAS FTP 網站下載 LinPAC SDK。

下表為 AM335x SDK 的 Modbus 測試工具檔案路徑，關於 LinPAC SDK 的更多下載資訊，請參考以下連結：<https://www.icpdas.com/en/download/show.php?num=1195>

作業系統	SDK 路徑
Windows	C:\cygwin\LinPAC_AM335x_SDK\examples\xvboard 或 C:\cygwin\LinPAC_AM335x_SDK\examples\modbus
Linux	root@LinuxPC-ICPDAS:/icpdas/linpac_am335x_sdk/i8k/examples/xvboard 或 root@LinuxPC-ICPDAS:/icpdas/linpac_am335x_sdk/i8k/examples/modbus

本例使用 LP-5231 控制器搭配 tM-DA1P1R1 與 tM-AD4P2C2 模組來做測試，以下為模組的設置參數：



用戶必須設置從站設備的設備站號 (Net ID) · 其設置步驟如下：

```
root@LP-5231:~# getmodbus 2 9600 1 4 484 1 100
1
root@LP-5231:~# setmodbus 2 9600 1 16 484 1 2 100
root@LP-5231:~# getmodbus 2 9600 2 4 484 1 100
2
```

步驟 1： 使用以下指令查詢 tM-AD4P2C2 模組的設備站號 (Net ID)。

指令：

```
getModbus <comport> <baudrate> <NetID> <command> <address> <count> <timeout(ms)>
```

```
# getModbus 2 9600 1 4 484 1 100
```

步驟 2： 使用以下指令修改 tM-AD4P2C2 模組的設備站號 (Net ID)。

指令：

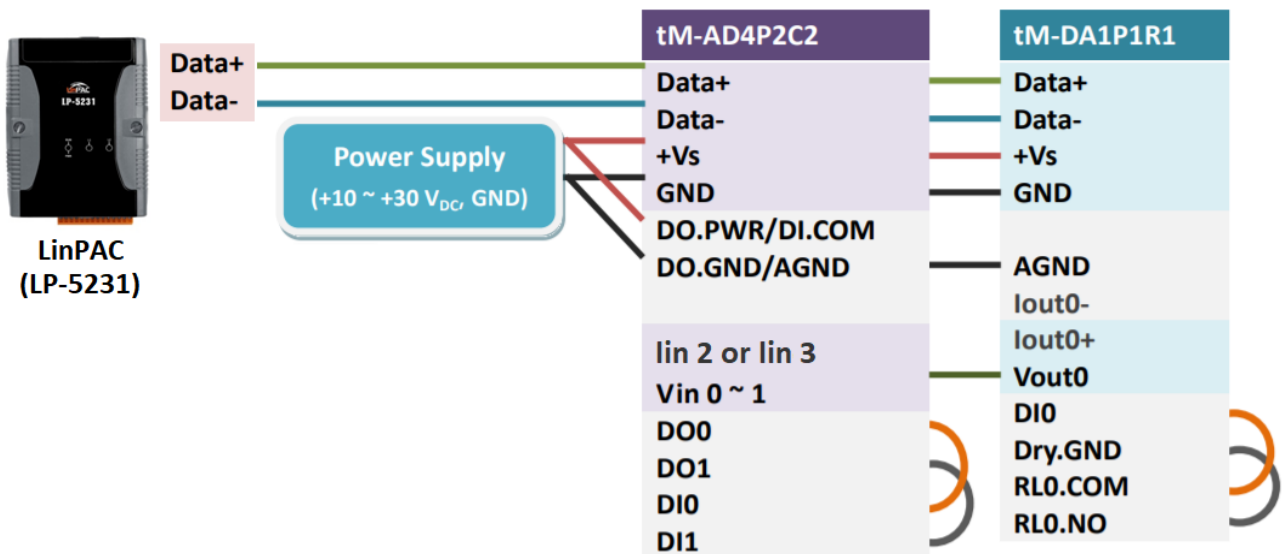
```
setModbus <comport> <baudrate> <NetID> <command> <address> <count> <value> <timeout(ms)>
```

```
# setModbus 2 9600 1 16 484 1 2 100 // Set the NetID as 2
```

注意：

1. 用戶可參考 附錄 C - Modbus 位址對應 (Base 1) · 來設置設備位址 (Net ID)。
2. tM 系列模組的基底位址 (base address) 是由 0 開始 (Base 0)。例如: Modbus 位址 400485 是用來讀取/寫入模組位址 (即 · Net ID) · 此例 · 需使用位址 484 來取得/設定 Net ID。

硬體接線 (測試用)：



□ 驗證 AI/AO 功能

連接 tM-DA1P1R1 模組的 AO 通道與 tM-AD4P2C2 模組的 AI 通道，使用範例程式 setModbus.c 與 getModbus.c 測試 AI/AO 功能。測試步驟如下：

```
root@LP-5231:~# setmodbus 2 9600 1 16 32 1 65535 100
root@LP-5231:~# getmodbus 2 9600 1 3 32 1 100
65535
root@LP-5231:~# getmodbus 2 9600 2 4 0 1 100
32767
```

步驟 1：使用以下指令，設定 tM-DA1P1R1 模組的 AO 值。

```
# setmodbus 2 9600 1 16 32 1 65535 100 //輸出 10V 電壓
```

步驟 2：使用以下指令，讀回 tM-DA1P1R1 模組的 AO 值。

```
# getmodbus 2 9600 1 3 32 1 100
```

步驟 3：使用以下指令，讀取 tM-AD4P2C2 模組的 AI 值。

```
# getmodbus 2 9600 2 4 0 1 100
```

□ 驗證 DI/DO 功能

連接 tM-DA1P1R1 模組的 DO 通道與 DI 通道，使用範例程式 setModbus.c 與 getModbus.c 測試 DI/DO 功能。測試步驟如下：

```
root@LP-5231:~# setmodbus 2 9600 1 15 0 1 1 100
wCount=1 iCount=8 iIndex=0
root@LP-5231:~# getmodbus 2 9600 1 1 0 1 100
1
root@LP-5231:~# getmodbus 2 9600 1 2 32 1 100
1
```

步驟 1：使用以下指令，設定 tM-DA1P1R1 模組的 DO 狀態。

```
# setmodbus 2 9600 1 15 0 1 1 100 //設置 DO 通道狀態為 ON
```

步驟 2：使用以下指令，讀回 tM-DA1P1R1 模組的 DO 狀態。

```
# getmodbus 2 9600 1 1 0 1 100
```


步驟 3：使用以下指令，讀取 tM-DA1P1R1 模組的 DI 狀態。

```
# getmodbus 2 9600 1 2 32 1 100
```

連接 tM-AD4P2C2 模組的 DO 通道與 DI 通道，使用範例程式 setModbus.c 與 getModbus.c 測試 DI/DO 功能。測試步驟如下：

```
root@LP-5231:~# setmodbus 2 9600 2 15 0 1 1 100
wCount=1 iCount=8 iIndex=0
root@LP-5231:~# getmodbus 2 9600 2 1 0 1 100
1
root@LP-5231:~# getmodbus 2 9600 2 2 32 1 100
1
```

步驟 1：使用以下指令，設定 tM-AD4P2C2 模組的 DO 狀態。

```
# setmodbus 2 9600 2 15 0 1 1 100 //設置 DO 通道狀態為 ON
```

步驟 2：使用以下指令，讀回 tM-AD4P2C2 模組的 DO 狀態。

```
# getmodbus 2 9600 2 1 0 1 100
```

步驟 3：使用以下指令，讀取 tM-AD4P2C2 模組的 DI 狀態。

```
# getmodbus 2 9600 2 2 32 1 100
```

2.1 LinPAC 與 Python 應用

LinPAC 系列產品支援 Python 語言的開發環境，用戶可在 LinPAC 中安裝 Python 官網提供的 Modbus 工具，來測試 tM 系列模組。本例使用 LP-5231 控制器搭配 tM-DA1P1R1 模組，應用 modbus-tk 工具來存取資料，關於 modbus-tk 軟體的更多資訊，請參考以下連結：

<https://github.com/ljean/modbus-tk>，其安裝與測試步驟如下：

步驟 1：使用以下指令，檢查 Python 的版本號，並確認 Python 為 2.5 以上版本。

```
# python --version
```

```
root@LP-5231:~# python --version
Python 2.7.3
root@LP-5231:~#
```

步驟 2：使用以下指令，安裝 pyserial 模組。

```
# pip install pyserial
```

步驟 3：使用以下指令，下載 modbus-tk 套件。

```
# wget https://github.com/ljean/modbus-tk/archive/master.zip
```

步驟 4：使用以下指令，解壓縮 modbus-tk 套件。

```
# unzip master.zip
```

步驟 5：使用以下指令，安裝 modbus-tk 工具。

```
# cd modbus-tk-master
# python setup.py build
# python setup.py install
```

步驟 6：使用以下指令，檢查 pyserial 與 modbus-tk 套件是否安裝成功。

```
# pip list
```

```
root@LP-5231:~/modbus-tk-master# pip list
Package      Version
-----
distribute  0.6.24dev-r0
modbus-tk    0.5.8
pip          18.1
pyserial     3.4
setuptools   0.6rc11
```

步驟 7：於 modbus-tk 套件所提供的範例中，找到範例程式 rtumaster_example.py。

```
root@LP-5231:~# cd modbus-tk-master/examples/
root@LP-5231:~/modbus-tk-master/examples# ls
modbus_system_monitor.py  rtumaster_example.py  tcpmaster_example.py
mysimu.py                 rtuslave_example.py  tcpslave_example.py
root@LP-5231:~/modbus-tk-master/examples#
```

步驟 8：修改範例程式 rtumaster_example.py 的參數設定。

```

import modbus_tk
import modbus_tk.defines as cst
from modbus_tk import modbus_rtu
#PORT = 1
PORT = '/dev/ttyO2' 通訊埠
def main():
    """main"""
    logger = modbus_tk.utils.create_logger("console")
    try:
        #Connect to the slave
        master = modbus_rtu.RtuMaster(
            serial.Serial(port=PORT, baudrate=9600, bytesize=8, parity='N', stopbits=1, xonxoff=0)
        )
        master.set_timeout(5.0)
        master.set_verbose(True)
        logger.info("connected")
        logger.info(master.execute(1, cst.READ_HOLDING_REGISTERS, 32, 1))
    #
    #
    #
    #

```

通訊埠參數設定

Modbus 命令
master.execute(NetID, function code, address, count)

注意： 在 LinPAC 系列設置 Modbus 時，Modbus 位址的起始位置由 0 開始。

步驟 9： 執行範例程式，來讀取 tM-DA1P1R1 模組的 AO 值，下圖為執行結果。

```

root@LP-5231:~/modbus-tk-master/examples# python rtumaster_example.py
2018-12-10 17:47:25,575 INFO modbus_rtu.__init__ MainThread RtuMaster /dev/ttyO2 is opened
2018-12-10 17:47:25,578 INFO rtumaster_example.main MainThread connected
2018-12-10 17:47:25,580 DEBUG modbus.execute MainThread -> 1-3-0-32-0-1-133-192
2018-12-10 17:47:25,606 DEBUG modbus.execute MainThread <- 1-3-2-255-255-185-244
2018-12-10 17:47:25,607 INFO rtumaster_example.main MainThread (65535,)

```

2.2 LinPAC 與 Perl 應用

LinPAC 系列產品支援 Perl 語言的開發環境，用戶可在 LinPAC 中安裝 Perl 官網提供的 Modbus 工具，來測試 tM 系列模組。本例使用 LP-5231 控制器搭配 tM-DA1P1R1 模組，應用 Device-Modbus-RTU 工具來存取資料，其安裝與測試步驟如下：

步驟 1： 下載並解壓縮 Device-Modbus-RTU 套件 (檔名: Device-Modbus-RTU-0.022.tar.gz)。

網址: <https://metacpan.org/release/Device-Modbus-RTU>

步驟 2：使用以下指令，安裝 Device-Modbus-RTU 的相依模組。

```
# sudo cpan Role::Tiny Try::Tiny Device::SerialPort Device::Modbus
```

步驟 3：使用以下指令，安裝 Device-Modbus-RTU 軟體。

```
# cd Device-Modbus-RTU-0.022
# perl Makefile.PL
# make
# make test
# make install
```

步驟 4：使用 'instModsh' 指令，檢查 perl 模組是否安裝成功。

```
root@LP-5231:~# instmodsh
Available commands are:
  l          - List all installed modules
  m <module> - Select a module
  q          - Quit the program
cmd? l
```

→

```
Installed modules are:
Device::Modbus
Device::Modbus::RTU
Device::SerialPort
Perl
Role::Tiny
Try::Tiny
cmd? q
```

步驟 5：於 Device-Modbus-RTU 套件所提供的範例中，找到範例程式 write_new_addr.pl 與 simple_client_rtu.pl。

```
root@LP-5231:~# cd Device-Modbus-RTU-master/examples/
root@LP-5231:~/Device-Modbus-RTU-master/examples# ls
arduino_client.ino  server_rtu.pl  simple_client_rtu.pl  write_new_addr.pl
```

步驟 6：修改範例程式的參數設定。

□ 修改 write_new_addr.pl 與 simple_client_rtu.pl 文本中的連接埠設定。

```
my $client = Device::Modbus::RTU::Client->new(
    port      => '/dev/ttyO2', // 通訊埠
    baudrate => 9600,          // 每秒傳輸位元
```

```
parity => 'none', // 同位檢查
);
```

❑ 修改 write_new_addr.pl 的 Modbus 命令設定，並設置 AO 通道的值。

```
my $req = $client->write_single_register( // Modbus 功能碼
    unit => 1, // 從站設備的站號 (NetID)
    address => 32, // 通道位址
    value => 65535 // 設定值
);
```

❑ 修改 simple_client_rtu.pl 的 Modbus 命令設定，並讀取 AO 通道的值。

```
my $req = $client->read_holding_registers( // Modbus 功能碼
    unit => 1, // 從站設備的站號 (NetID)
    address => 32, // 通道位址
    quantity => 1, // 所要操作的暫存器長度
);
```

範例程式 write_new_addr.pl

```
#!/usr/bin/env perl
use Device::Modbus;
use Device::Modbus::RTU::Client;
use Data::Dumper;
use strict;
use warnings;
use v5.10;
```

```
my $client = Device::Modbus::RTU::Client->new(
    port => '/dev/ttyO2',
    baudrate => 9600,
    parity => 'none',
);
```

通訊埠參數設定

```
my $req = $client->write_single_register(
    unit => 1,
    address => 32,
    value => 65535
);
```

Modbus 命令設定

```
say "->" . Dumper $req;
$client->send_request($req);
my $resp = $client->receive_response;
say "<-" . Dumper $resp;
```

注意： 在 LinPAC 系列設置 Modbus 時，Modbus 位址的起始位置由 0 開始。

步驟 7：執行範例程式來控制 tM 系列模組。

(1) 執行 write_new_addr.pl 的結果：



```

root@LP-5231:~/Device-Modbus-RTU-master/examples# perl write_new_addr.pl
->$VAR1 = bless( {
    'unit' => 1,
    'function' => 'Write Single Register',
    'value' => 65535,
    'address' => 32,
    'code' => 6
  }, 'Device::Modbus::Request' );
<-$VAR1 = bless( {
    'unit' => 1,
    'crc' => 45193,
    'message' => bless( {
        'function' => 'Write Single Register',
        'value' => 65535,
        'address' => 32,
        'code' => 6
      }, 'Device::Modbus::Response' )
  }, 'Device::Modbus::RTU::ADU' );

```

(2) 執行 simple_client_rtu.pl 的結果:

```

root@LP-5231:~/Device-Modbus-RTU-master/examples# perl simple_client_rtu.pl
->$VAR1 = bless( {
    'unit' => 1,
    'function' => 'Read Holding Registers',
    'quantity' => 1,
    'address' => 32,
    'code' => 3
  }, 'Device::Modbus::Request' );
<-$VAR1 = bless( {
    'unit' => 1,
    'crc' => 62649,
    'message' => bless( {
        'bytes' => 2,
        'function' => 'Read Holding Registers',
        'values' => [
            65535
        ],
        'code' => 3
      }, 'Device::Modbus::Response' )
  }, 'Device::Modbus::RTU::ADU' );

```

附錄 A. 如何在 Windows 平台上編譯包含 libmodbus 函式庫的應用程式



下載與安裝

在 Windows 下編譯 Libmodbus，使用者需要安裝 MinGW 和 MSYS，並可選用相關軟體包 (gcc、automake、libtool 等)，如下所示：

- **Libmodbus:** <https://github.com/stephane/libmodbus/releases>
- **MinGW for Windows Platform:** <https://sourceforge.net/projects/mingw/>
- **MSYS for Windows Platform :** <https://www.msys2.org/>

設環境變數

PATH 變數，定義了編譯程式會運行的搜尋與執行路徑。因此，使用者需要修改 `C:\msys64\etc\profile` 文件，將交叉編譯資料夾，加入到環境變數「PATH」的「MSYS」變數內。

以下以 **LP-8x21** 為例：

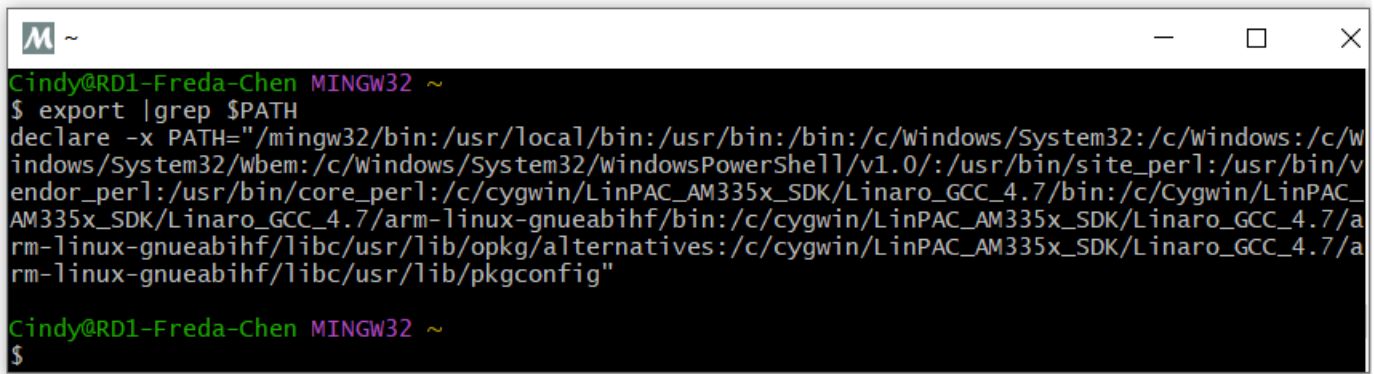
```
PATH=$PATH:/c/cygwin/LinPAC_AM335x_SDK/Linaro_GCC_4.7/bin:/c/Cygwin/LinPAC_AM335x_SDK/Linaro_GCC_4.7/arm-linux-gnueabi/hf/bin:/c/cygwin/LinPAC_AM335x_SDK/Linaro_GCC_4.7/arm-linux-gnueabi/hf/libc/usr/lib/opkg/alternatives:/c/cygwin/LinPAC_AM335x_SDK/Linaro_GCC_4.7/arm-linux-gnueabi/hf/libc/usr/lib/pkgconfig'
```

開啟 mingw32.exe shell 啟動器 (點選：「開始」功能表 → 「MSYS2」 → 「MSYS2 MINGW32」)



或進入 C:\msys64\目錄，點選 **mingw32.exe**

使用 '**export**' 指令，查看 PATH 環境變數是否已正確設置。



```
Cindy@RD1-Freda-Chen MINGW32 ~
$ export | grep $PATH
declare -x PATH="/mingw32/bin:/usr/local/bin:/usr/bin:/bin:/c/windows/System32:/c/windows/System32/wbem:/c/windows/System32/windowsPowerShell/v1.0/;/usr/bin/site_perl:/usr/bin/vendor_perl:/usr/bin/core_perl:/c/cygwin/LinPAC_AM335x_SDK/Linaro_GCC_4.7/bin:/c/Cygwin/LinPAC_AM335x_SDK/Linaro_GCC_4.7/arm-linux-gnueabi/hf/bin:/c/cygwin/LinPAC_AM335x_SDK/Linaro_GCC_4.7/arm-linux-gnueabi/hf/libc/usr/lib/opkg/alternatives:/c/cygwin/LinPAC_AM335x_SDK/Linaro_GCC_4.7/arm-linux-gnueabi/hf/libc/usr/lib/pkgconfig"
Cindy@RD1-Freda-Chen MINGW32 ~
$
```

更新 **MSYS2** 及安裝套件，如下：

- I. 安裝：pacman -Syu
- II. 安裝：pacman -Su
- III. 安裝：pacman -S autoconf-wrapper
- IV. 安裝：pacman -S mingw-w64-i686-toolchain
- V. 安裝：pacman -S autotools


```

M /c/cygwin/LinPAC_AM335x_SDK/libmodbus-3.1.10
Cindy@RD1-Freda-Chen MINGW32 /c/cygwin/LinPAC_AM335x_SDK/libmodbus-3.1.10
$ pacman -S autotools
resolving dependencies...
looking for conflicting packages...

Packages (10) automake-wrapper-11-4 automake1.11-1.11.6-6 automake1.12-1.12.6-6
              automake1.13-1.13.4-7 automake1.14-1.14.1-6 automake1.15-1.15.1-4
              automake1.16-1.16.5-1 libltdl-2.4.6-14 libtool-2.4.6-14 autotools-2022.01.16-2

Total Download Size: 3.38 MiB
Total Installed Size: 10.98 MiB

:: Proceed with installation? [Y/n] y
:: Retrieving packages...
automake1.15-1.15.1-4-any 513.4 KiB 86.5 KiB/s 00:06 [#####] 100%
automake1.16-1.16.5-1-any 526.3 KiB 86.4 KiB/s 00:06 [#####] 100%
automake1.12-1.12.6-6-any 503.1 KiB 75.6 KiB/s 00:07 [#####] 100%
automake1.11-1.11.6-6-any 490.2 KiB 341 KiB/s 00:01 [#####] 100%
libltdl-2.4.6-14-x86_64 32.0 KiB 23.5 KiB/s 00:01 [#####] 100%
libtool-2.4.6-14-x86_64 388.1 KiB 235 KiB/s 00:02 [#####] 100%
automake1.13-1.13.4-7-any 501.5 KiB 61.1 KiB/s 00:08 [#####] 100%
automake1.14-1.14.1-6-any 503.1 KiB 60.5 KiB/s 00:08 [#####] 100%
automake-wrapper-11-4-any 4.4 KiB 1210 B/s 00:04 [#####] 100%
autotools-2022.01.16-2-any 2.5 KiB 651 B/s 00:04 [#####] 100%
Total (10/10) 3.4 MiB 261 KiB/s 00:13 [#####] 100%
(10/10) checking keys in keyring [#####] 100%
(10/10) checking package integrity [#####] 100%
(10/10) loading package files [#####] 100%
(10/10) checking for file conflicts [#####] 100%
(10/10) checking available disk space [#####] 100%
:: Processing package changes...
( 1/10) installing automake1.11 [#####] 100%
( 2/10) installing automake1.12 [#####] 100%
( 3/10) installing automake1.13 [#####] 100%
( 4/10) installing automake1.14 [#####] 100%
( 5/10) installing automake1.15 [#####] 100%
( 6/10) installing automake1.16 [#####] 100%
( 7/10) installing automake-wrapper [#####] 100%
( 8/10) installing libltdl [#####] 100%
( 9/10) installing libtool [#####] 100%
(10/10) installing autotools [#####] 100%
:: Running post-transaction hooks...
(1/1) Updating the info directory file...

```

輸入 "gcc -v" 指令，檢查 gcc 版本。

```

M ~
Cindy@RD1-Freda-Chen MINGW32 ~
$ gcc -v
Using built-in specs.
COLLECT_GCC=C:/msys64/mingw32/bin/gcc.exe
COLLECT_LTO_WRAPPER=C:/msys64/mingw32/bin/./lib/gcc/i686-w64-mingw32/12.2.0/lto-wrapper.exe
Target: i686-w64-mingw32
Configured with: ../gcc-12.2.0/configure --prefix=/mingw32 --with-local-prefix=/mingw32/local --build=i686-w64-mingw32 --host=i686-w64-mingw32 --target=i686-w64-mingw32 --with-native-system-header-dir=/mingw32/include --libexecdir=/mingw32/lib --enable-bootstrap --enable-checking=release --with-arch=i686 --with-tune=generic --enable-languages=c,lto,c++,fortran,ada,objc,obj-c++,jit --enable-shared --enable-static --enable-libatomic --enable-threads=posix --enable-graphite --enable-fully-dynamic-string --enable-libstdcxx-fs-backend=fts --enable-libstdcxx-time --disable-libstdcxx-pch --enable-lto --enable-libgomp --disable-multilib --disable-rpath --disable-win32-registry --disable-nls --disable-werror --disable-symvers --with-libiconv --with-system-zlib --with-gmp=/mingw32 --with-mpfr=/mingw32 --with-mpc=/mingw32 --with-isl=/mingw32 --with-pkgversion='Rev4, Built by MSYS2 project' --with-bugurl=https://github.com/msys2/MINGW-packages/issues --with-gnu-as --with-gnu-ld --disable-libstdcxx-debug --disable-sjlj-exceptions --with-dwarf2 --with-boot-ldflags=-static-libstdc++ --with-stage1-ldflags=-static-libstdc++
Thread model: posix
Supported LTO compression algorithms: zlib zstd
gcc version 12.2.0 (Rev4, Built by MSYS2 project)

Cindy@RD1-Freda-Chen MINGW32 ~
$

```

輸入 'arm-linux-gnueabi-gcc -v' 指令，以檢查交叉編譯器是否予許採用。

```
Cindy@RD1-Freda-Chen MINGW32 ~
$ arm-linux-gnueabi-gcc -v
 specs
COLLECT_GCC=C:\cygwin\LinPAC_AM335x_SDK\Linaro_GCC_4.7\bin\arm-linux-gnueabi-gcc.exe
COLLECT_LTO_WRAPPER=c:/cygwin/linpac_am335x_sdk/linaro_gcc_4.7/bin/./libexec/gcc/arm-linux-gnueabi/4.7.3/lto-w
rapper.exe
arm-linux-gnueabi
/cbuild/slaves/oorts/crostoool-ng/builds/arm-linux-gnueabi-win32/.build/src/gcc-linaro-4.7-2013.03/configure --
build=i686-build_pc-linux-gnu --host=i586-host_pc-mingw32msvc --target=arm-linux-gnueabi --prefix=/cbuild/slave
s/oorts/crostoool-ng/builds/arm-linux-gnueabi-win32/install --with-sysroot=/cbuild/slaves/oorts/crostoool-ng/bu
ilds/arm-linux-gnueabi-win32/install/arm-linux-gnueabi/libc --enable-languages=c,c++,fortran --enable-multili
b --with-arch=armv7-a --with-tune=cortex-a9 --with-fpu=vfpv3-d16 --with-float=hard --with-pkgversion='crostoool-N
G linaro-1.13.1-4.7-2013.03-20130313 - Linaro GCC 2013.03' --with-bugurl=https://bugs.launchpad.net/gcc-linaro --
enable-__cxa_atexit --enable-libmudflap --enable-libgomp --enable-libssp --with-gmp=/cbuild/slaves/oorts/crosto
ool-ng/builds/arm-linux-gnueabi-win32/.build/arm-linux-gnueabi/build/static --with-mpfr=/cbuild/slaves/oorts/cr
osstoool-ng/builds/arm-linux-gnueabi-win32/.build/arm-linux-gnueabi/build/static --with-mpc=/cbuild/slaves/oor
ts/crostoool-ng/builds/arm-linux-gnueabi-win32/.build/arm-linux-gnueabi/build/static --with-pp1=/cbuild/slave
s/oorts/crostoool-ng/builds/arm-linux-gnueabi-win32/.build/arm-linux-gnueabi/build/static --with-cloog=/cbuil
d/slaves/oorts/crostoool-ng/builds/arm-linux-gnueabi-win32/.build/arm-linux-gnueabi/build/static --with-libel
f=/cbuild/slaves/oorts/crostoool-ng/builds/arm-linux-gnueabi-win32/.build/arm-linux-gnueabi/build/static --wi
th-host-libstdcxx='-L/cbuild/slaves/oorts/crostoool-ng/builds/arm-linux-gnueabi-win32/.build/arm-linux-gnueabi
f/build/static/lib -lpwl' --enable-threads=posix --disable-libstdcxx-pch --enable-linker-build-id --enable-gold -
-with-local-prefix=/cbuild/slaves/oorts/crostoool-ng/builds/arm-linux-gnueabi-win32/install/arm-linux-gnueabi
/libc --enable-c99 --enable-long-long --with-mode=thumb
posix
gcc version 4.7.3 20130226 (prerelease) (crostoool-NG linaro-1.13.1-4.7-2013.03-20130313 - Linaro GCC 2013.03)

Cindy@RD1-Freda-Chen MINGW32 ~
$
```

MSYS2 Packages : <https://packages.msys2.org/groups/mingw-w64-i686-toolchain>

編譯 LIBMODBUS 原始碼及其範例

首先執行 “`sh autogen.sh`” 以產生設定腳本。

```
M /c/cygwin/LinPAC_AM335x_SDK/libmodbus-3.1.10
Cindy@RD1-Freda-Chen MINGW32 /c/cygwin/LinPAC_AM335x_SDK/libmodbus-3.1.10
$ sh autogen.sh
libtoolize: putting auxiliary files in AC_CONFIG_AUX_DIR, 'build-aux'.
libtoolize: linking file 'build-aux/ltmain.sh'
libtoolize: putting macros in AC_CONFIG_MACRO_DIRS, 'm4'.
libtoolize: linking file 'm4/libtool.m4'
libtoolize: linking file 'm4/ltoptions.m4'
libtoolize: linking file 'm4/ltugar.m4'
libtoolize: linking file 'm4/ltversion.m4'
libtoolize: linking file 'm4/lt~obsolete.m4'
configure.ac:33: installing 'build-aux/compile'
configure.ac:56: installing 'build-aux/config.guess'
configure.ac:56: installing 'build-aux/config.sub'
configure.ac:32: installing 'build-aux/install-sh'
configure.ac:32: installing 'build-aux/missing'
src/Makefile.am: installing 'build-aux/depcomp'
parallel-tests: installing 'build-aux/test-driver'

-----
Initialized build system. You can now run ./configure
-----
```

若要進入 `libmodbus-3.1.8` 目錄，可輸入：

```
Cindy@RD1-Freda-Chen MINGW32 /c/cygwin/LinPAC_AM335x_SDK/libmodbus-3.1.8
$ mkdir linpac

Cindy@RD1-Freda-Chen MINGW32 /c/cygwin/LinPAC_AM335x_SDK/libmodbus-3.1.8
$ sh configure CC=arm-linux-gnueabi-gcc --host=arm-linux-gnueabi --enable-static
--prefix=$(pwd)/linpac
```

使用 “`make`” 指令來編譯程式。

```
M /c/cygwin/LinPAC_AM335x_SDK/libmodbus-3.1.10
Cindy@RD1-Freda-Chen MINGW32 /c/cygwin/LinPAC_AM335x_SDK/libmodbus-3.1.10
$ make
make --no-print-directory all-recursive
Making all in src
CC      modbus.lo
CC      modbus-data.lo
CC      modbus-rtu.lo
CC      modbus-tcp.lo
CCLD   libmodbus.la
Making all in tests
make all-am
CC      bandwidth-server-one.o
CCLD   bandwidth-server-one
CC      bandwidth-server-many-up.o
CCLD   bandwidth-server-many-up
CC      bandwidth-client.o
CCLD   bandwidth-client
CC      random-test-server.o
CCLD   random-test-server
CC      random-test-client.o
CCLD   random-test-client
CC      unit-test-server.o
CCLD   unit-test-server
CC      unit-test-client.o
CCLD   unit-test-client
CC      version.o
CCLD   version
```

使用 “make install” 命令安裝 Modbus 函式庫及其表頭檔：

```
M /c/cygwin/LinPAC_AM335x_SDK/libmodbus-3.1.10
Cindy@RD1-Freda-Chen MINGW32 /c/cygwin/LinPAC_AM335x_SDK/libmodbus-3.1.10
$ make install
Making install in src
/usr/bin/mkdir -p '/c/cygwin/LinPAC_AM335x_SDK/libmodbus-3.1.10/linpac/lib'
/bin/sh ../libtool --mode=install /usr/bin/install -c libmodbus.la '/c/cygwin/LinPAC_AM335x_SDK/libmodbus-3.1.10/linpac/lib'
libtool: install: /usr/bin/install -c .libs/libmodbus.so.5.1.0 /c/cygwin/LinPAC_AM335x_SDK/libmodbus-3.1.10/linpac/lib/libmodbus.so.5.1.0
libtool: install: (cd /c/cygwin/LinPAC_AM335x_SDK/libmodbus-3.1.10/linpac/lib && { cp -pR -f libmodbus.so.5.1.0 libmodbus.so.5 || { rm -f libmodbus.so.5 && cp -pR libmodbus.so.5.1.0 libmodbus.so.5; }; })
libtool: install: (cd /c/cygwin/LinPAC_AM335x_SDK/libmodbus-3.1.10/linpac/lib && { cp -pR -f libmodbus.so.5.1.0 libmodbus.so || { rm -f libmodbus.so && cp -pR libmodbus.so.5.1.0 libmodbus.so; }; })
libtool: install: /usr/bin/install -c .libs/libmodbus.lai /c/cygwin/LinPAC_AM335x_SDK/libmodbus-3.1.10/linpac/lib/libmodbus.la
libtool: install: /usr/bin/install -c .libs/libmodbus.a /c/cygwin/LinPAC_AM335x_SDK/libmodbus-3.1.10/linpac/lib/libmodbus.a
libtool: install: chmod 644 /c/cygwin/LinPAC_AM335x_SDK/libmodbus-3.1.10/linpac/lib/libmodbus.a
libtool: install: arm-linux-gnueabi-hf-ranlib /c/cygwin/LinPAC_AM335x_SDK/libmodbus-3.1.10/linpac/lib/libmodbus.a
libtool: finish: PATH="/mingw32/bin:/usr/local/bin:/usr/bin:/bin:/c/Windows/System32:/c/Windows:/c/Windows/System32/Wbem:/c/Windows/System32/WindowsPowerShell/v1.0:/usr/bin/site_perl:/usr/bin/vendor_perl:/usr/bin/core_perl:/c/cygwin/LinPAC_AM335x_SDK/Linaro_GCC_4.7/bin:/c/cygwin/LinPAC_AM335x_SDK/Linaro_GCC_4.7/arm-linux-gnueabi-hf/bin:/c/cygwin/LinPAC_AM335x_SDK/Linaro_GCC_4.7/arm-linux-gnueabi-hf/libc/usr/lib/opkg/alternatives:/c/cygwin/LinPAC_AM335x_SDK/Linaro_GCC_4.7/arm-linux-gnueabi-hf/libc/usr/lib/pkgconfig:/sbin" ldconfig -n /c/cygwin/LinPAC_AM335x_SDK/libmodbus-3.1.10/linpac/lib
/c/cygwin/LinPAC_AM335x_SDK/libmodbus-3.1.10/libtool: line 1737: ldconfig: command not found

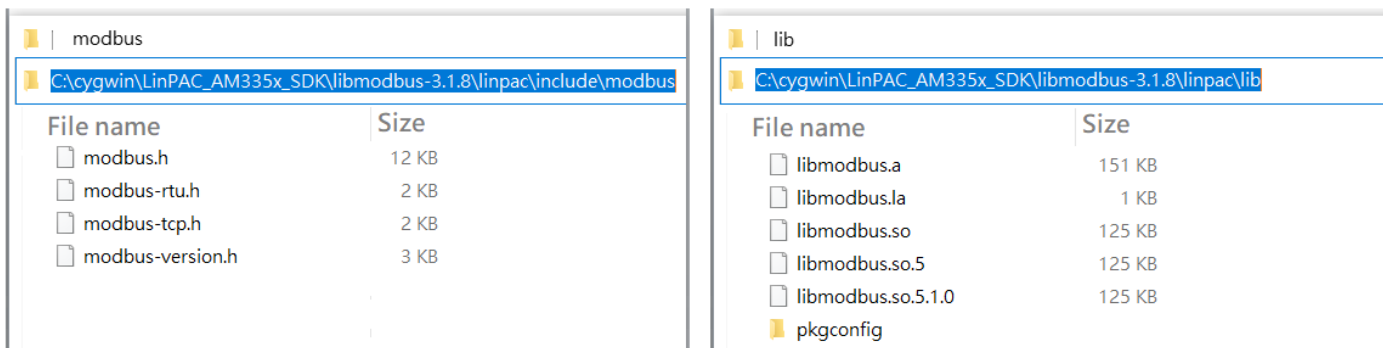
-----
Libraries have been installed in:
  /c/cygwin/LinPAC_AM335x_SDK/libmodbus-3.1.10/linpac/lib

If you ever happen to want to link against installed libraries
in a given directory, LIBDIR, you must either use libtool, and
specify the full pathname of the library, or use the '-LLIBDIR'
flag during linking and do at least one of the following:
- add LIBDIR to the 'LD_LIBRARY_PATH' environment variable
  during execution
- add LIBDIR to the 'LD_RUN_PATH' environment variable
  during linking
- use the '-Wl,-rpath -Wl,LIBDIR' linker flag
- have your system administrator run these commands:

See any operating system documentation about shared libraries for
more information, such as the ld(1) and ld.so(8) manual pages.

-----
/usr/bin/mkdir -p '/c/cygwin/LinPAC_AM335x_SDK/libmodbus-3.1.10/linpac/include/modbus'
/usr/bin/install -c -m 644 modbus.h modbus-version.h modbus-rtu.h modbus-tcp.h '/c/cygwin/LinPAC_AM335x_SDK/libmodbus-3.1.10/linpac/include/modbus'
Making install in tests
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Nothing to be done for 'install-exec-am'.
/usr/bin/mkdir -p '/c/cygwin/LinPAC_AM335x_SDK/libmodbus-3.1.10/linpac/share/doc/libmodbus'
/usr/bin/install -c -m 644 AUTHORS NEWS README.md '/c/cygwin/LinPAC_AM335x_SDK/libmodbus-3.1.10/linpac/share/doc/libmodbus'
/usr/bin/mkdir -p '/c/cygwin/LinPAC_AM335x_SDK/libmodbus-3.1.10/linpac/lib/pkgconfig'
/usr/bin/install -c -m 644 libmodbus.pc '/c/cygwin/LinPAC_AM335x_SDK/libmodbus-3.1.10/linpac/lib/pkgconfig'
```

以下是在 Windows 平台上，完整安裝 libmodbus-3.1.8 的結果：



手動編譯 modbus 範例，指令如下：

```
Cindy@RD1-Freda-Chen MINGW32 /c/cygwin/LinPAC_AM335x_SDK/libmodbus-3.1.8/tests
$ arm-linux-gnueabi-gcc -I../linpac/include/modbus -lm -o random-test-server.exe
random-test-server.c ../linpac/lib/libmodbus.a
```

使用者可自行修改 **Makefile** 檔，添加以下內容：

- LDFLAGS = -lm
- CFLAGS = -g -O2 -I. -I../include
- LIBS = ../linpac/lib/libmodbus.a
- Change syntax of a makefile's contents --- 'version' for example.

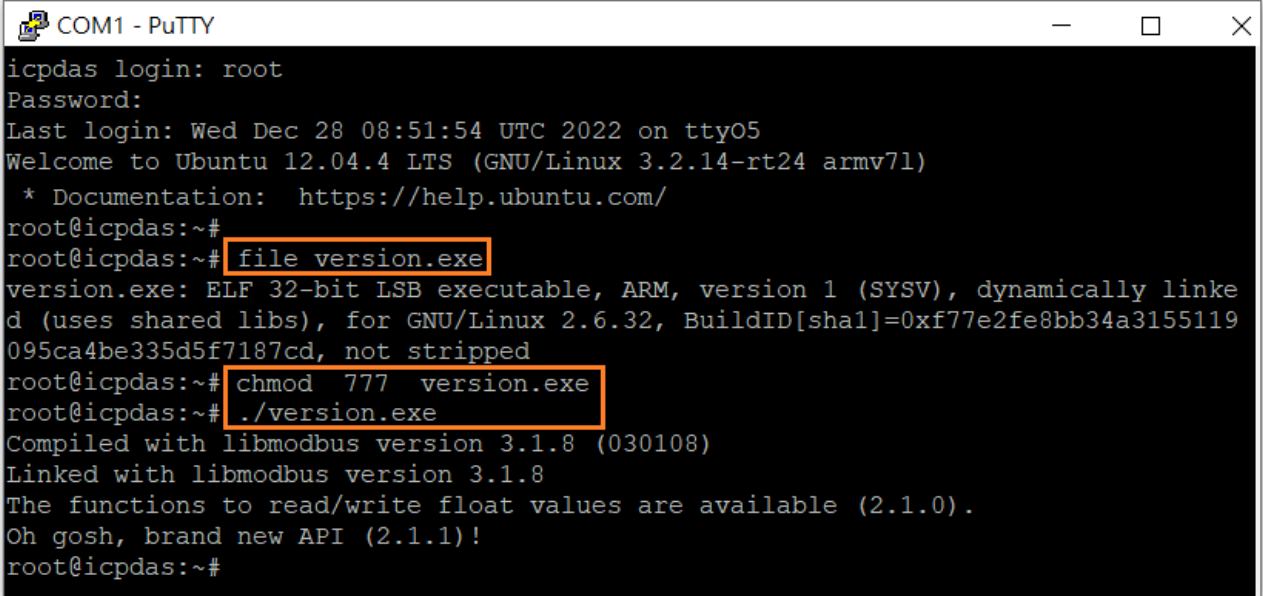
```
676 #version$(EXEEXT): $(version_OBJECTS) $(version_DEPENDENCIES) $(EXTRA_version_DEPENDENCIES)
677 # @rm -f version$(EXEEXT)
678 # $(AM_V_CCLD)$ (LINK) $(version_OBJECTS) $(version_LDADD) $(LIBS)
679
680 version: ./version.o
681     $(CC) $(CFLAGS) -o ./.$@ ./version.o $(LIBS) $(LDFLAGS)
682     @rm -f ./version.o
```

```
M /c/cygwin/LinPAC_AM335x_SDK/libmodbus-3.1.8/tests
Cindy@RD1-Freda-Chen MINGW32 /c/cygwin/LinPAC_AM335x_SDK/libmodbus-3.1.8/tests
$ make clean
test -z "%~ *.log" || rm -f %~ *.log
mrm -rf .libs _libs
ak rm -f bandwidth-server-one bandwidth-server-many-up bandwidth-client random-test-server random-test-client unit-test-server unit-test-client version
rm -f *.o
etest -z "/unit-tests.sh.log" || rm -f ./unit-tests.sh.log
test -z "/unit-tests.sh.trs" || rm -f ./unit-tests.sh.trs
test -z "test-suite.log" || rm -f test-suite.log
rm -f *.lo

Cindy@RD1-Freda-Chen MINGW32 /c/cygwin/LinPAC_AM335x_SDK/libmodbus-3.1.8/tests
$ make
make all-am
make[1]: Entering directory '/c/cygwin/LinPAC_AM335x_SDK/libmodbus-3.1.8/tests'
CC      bandwidth-server-one.o
CCLD   bandwidth-server-one
CC      bandwidth-server-many-up.o
CCLD   bandwidth-server-many-up
CC      bandwidth-client.o
CCLD   bandwidth-client
CC      random-test-server.o
CCLD   random-test-server
CC      random-test-client.o
CCLD   random-test-client
CC      unit-test-server.o
CCLD   unit-test-server
CC      unit-test-client.o
CCLD   unit-test-client
CC      version.o
arm-linux-gnueabi-gcc -std=gnu11 -g -O2 -I. -I../include -o ./version ./version.o ../linpac/lib/libmodbus.a -lm
make[1]: Leaving directory '/c/cygwin/LinPAC_AM335x_SDK/libmodbus-3.1.8/tests'

Cindy@RD1-Freda-Chen MINGW32 /c/cygwin/LinPAC_AM335x_SDK/libmodbus-3.1.8/tests
$ file version
version: ELF 32-bit LSB executable, ARM, EABI5 version 1 (SYSV), dynamically linked, interpreter /lib/ld-linux-armhf.so.3, for GNU/Linux 2.6.32, BuildID[sha1]=32bff971c50d858f27f93171b33a29bbfcb2bc, with debug_info, not stripped
Cindy@RD1-Freda-Chen MINGW32 /c/cygwin/LinPAC_AM335x_SDK/libmodbus-3.1.8/tests
$
```

上傳執行檔至 LINPAC



```
COM1 - PuTTY
icpdas login: root
Password:
Last login: Wed Dec 28 08:51:54 UTC 2022 on ttyO5
Welcome to Ubuntu 12.04.4 LTS (GNU/Linux 3.2.14-rt24 armv7l)
 * Documentation:  https://help.ubuntu.com/
root@icpdas:~#
root@icpdas:~# file version.exe
version.exe: ELF 32-bit LSB executable, ARM, version 1 (SYSV), dynamically linke
d (uses shared libs), for GNU/Linux 2.6.32, BuildID[sha1]=0xf77e2fe8bb34a3155119
095ca4be335d5f7187cd, not stripped
root@icpdas:~# chmod 777 version.exe
root@icpdas:~# ./version.exe
Compiled with libmodbus version 3.1.8 (030108)
Linked with libmodbus version 3.1.8
The functions to read/write float values are available (2.1.0).
Oh gosh, brand new API (2.1.1)!
root@icpdas:~#
```

附錄 B. 修訂記錄

本章提供本文檔的修訂歷史資訊。

下表顯示了修訂記錄：

版本	日期	內容
V1.0.0	May 2023	Initial issue
V1.0.1	March 2024	新增: How to compile application including libmodbus library on Windows Platform?