

# Modbus Development kit for Linux



# V1.0.2 Oct. 2024 Edited by Cindy Huang

Modbus Development kit for Linux

版本:1.0.2

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 C 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

Modbus Development kit for Linux

# **Contents**

1.	開女	始入門	4
	1.1.	Linux SDK	4
	1.2	. 下載 LinPAC SDK	5
2.	Linu	ux Modbus 開發工具	6
	2.1	LinPAC 與 C 應用	6
		2.1.1 LinPAC SDK	5 0
	2.1	LinPAC 與 Python 應用1	2
	2.2	LinPAC 與 Perl 應用14	4
附錄	ξA.	如何在 Windows 平台上編譯包含 libmodbus 函式庫的應用程式1	7
附錄	ξ B.	修訂記錄	5

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 。

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

目前 LinPAC SDK 的編譯操作方式,依產品分做二大類:

# 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	LP-224x LP-52xx	https://www.icpdas.com/en/download/show.php?num=1195&model=LP-5231M
Series	LP-8x2x LP-9x2x	https://www.icpdas.com/en/download/show.php?num=915&model=LP-9821
X86/E38xx Series	LX-Series	https://www.icpdas.com/en/download/show.php?num=904&model=LX-9371

# ▲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 .e	xe檔至C槽	
Windows	C: \cygwin\LinPAC_AM335x_SDK\examples\ <b>xvboard\</b> 或 C: \cygwin\LinPAC_AM335x_SDK\examples\ <b>modbus\</b>	
解壓縮/安裝.bz2 檔至 root (/) 目錄		
Linux root@LinuxPC-ICPDAS: /icpdas/linpac_am335x_sdk/i8k/examples/xv root@LinuxPC-ICPDAS: /icpdas/linpac_am335x_sdk/i8k/examples/m		

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。

Modbus 開發工具	下載方式
C 語言	
<ul><li>★ LinPAC SDK</li><li>★ libmodbus</li></ul>	至 ICP DAS 官網下載 LinPAC SDK 至 libmodbus 官網下載 libmodbus library
Python 語言	
★ modbus-tk	至 Python 官網下載 Modbus 工具
Perl 語言	
★ Device-Modbus	至 Perl 官網下載 Modbus 工具

# 2.1 LinPAC 與 C 應用

## 2.1.1 LinPAC SDK

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: <u>https://www.icpdas.com/en/download/show.php?num=1195</u>

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

Modbus Development kit for Linux

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:

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)。

<u>指令</u>:

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

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

執行結果:

```
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
```

<u>注意</u>:

- 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。

Modbus Development kit for Linux	版本:1.0.2
----------------------------------	----------



#### □ 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:

步驟1:使用以下指令,設定 tM-DA1P1R1 模組的 AO 值。

電壓
ŧ

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

執行結果:

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

Modbus Development kit for Linux

#### □ 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:

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

執行結果如下:

```
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
```

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:

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

執行結果如下:

```
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
```

Modbus Development kit for Linux

```
版本:1.0.2
```

版權所有 © 2023 ICP DAS Co., Ltd. All Rights Reserved.

第9頁

## 2.1.2 libmodbus

本例使用 LP-8421 控制器搭配 M-7060 模組來做測試,以下為模組的設置參數:



步驟1:至官網下載LinPAC SDK於Linux PC,並設定好LinPAC SDK環境變數

指令:

#. /icpdas/linpac\_am335x\_sdk/linpac\_am335x.sh

**步驟 2:**下載 libmodbus 原始碼 (<u>https://libmodbus.org/</u>) 至 Linux PC 端,並執行編譯 與安裝 指令:

# ./configure --host=arm-linux-gnueabihf --enable-static --prefix=/pj/tools/libmodbus-3.1.10/linpac # make

# make install

🚰 cindy@ICPDAS: /pj/tools/libmodbus-3.1.10/linpac — 🗆 🗙							
cindy@ICPDAS:/pj/too	ols/libmodbus-3.	.1.10\$ ls					
aclocal.m4	config.h	CONTRIBUTING.md	libtool	mkdocs.yml	tests		
AUTHORS	config.h.in	COPYING.LESSER	linpac	NEWS			
autogen.sh	config.log	docs	m4	README.md			
autom4te.cache	config.status	ISSUE_TEMPLATE.md	Makefile	SECURITY.md			
build-aux	configure	libmodbus.pc	Makefile.am	src			
CODE_OF_CONDUCT.md	configure.ac	libmodbus.pc.in	Makefile.in	stamp-h1			
cindy@ICPDAS:/pj/too	cindy@ICPDAS:/pj/tools/libmodbus-3.1.10\$ cd linpac/						
cindy@ICPDAS:/pj/too	cindy@ICPDAS:/pj/tools/libmodbus-3.1.10/linpac\$ ls						
include lib share							
cindy@ICPDAS:/pj/tools/libmodbus-3.1.10/linpac\$ ls lib							
libmodbus.a libmodbus.la libmodbus.so libmodbus.so.5 libmodbus.so.5.1.0 pkgconfig							
cindy@ICPDAS:/pj/too	ols/libmodbus-3.	.1.10/linpac\$					

註:若發生編譯錯誤,可於 Linux PC 安裝: apt-get install lib32z1

Modbus Development kit for Linux

第10頁

步驟 3:編寫程式(或參考 tests 目錄下範例做修改)、進行編譯。完成後,將執行檔上傳至 LinPAC

如下圖:

Proot@ICPDAS: /pj/tools/libmodbus-3.1.10/tests	_		×
root@ICPDAS:/pj/tools/libmodbus-3.1.10/tests# vi hello-rtu.c root@ICPDAS:/pj/tools/libmodbus-3.1.10/tests# arm-linux-gnueabihf-gcc -I/linpa	c/includ	e/modb	ous -
lm -o hello-rtu hello-rtu.c/linpac/lib/libmodbus.a			
root@ICPDAS:/pj/tools/libmodbus-3.1.10/tests# ftp 10.1.0.36			
Connected to 10.1.0.36.			
220 (vsFTPd 2.3.5)			
Name (10.1.0.36:cindy): root			
331 Please specify the password.			
Password:			
230 Login successful.			
Remote system type is UNIX.			
Using binary mode to transfer files.			
ftp> bin			
200 Switching to Binary mode.			
ftp> put hello-rtu			
local: hello-rtu remote: hello-rtu			
229 Entering Extended Passive Mode (   49032 ).			
150 Ok to send data.			
100%  ***********************************	iB/s	00:00	ETA
226 Transfer complete.			
41811 bytes sent in 00:00 (6.24 MiB/s)			
ftp> by			
221 Goodbye.			
rootATCPDAS./ni/tools/libmodbus_2 1 10/tests#			

```
cindy@ICPDAS: /pj/tools/libmodbus-3.1.10/tests
                                                                                      cindy@ICPDAS:/pj/tools/libmodbus-3.1.10/tests$ cat hello-rtu.c
#include <stdio.h>
#include <modbus.h>
#include <errno.h>
int main(void) {
int i, num;
modbus t *ctx = modbus new rtu("/dev/ttys0", 115200, 'N', 8, 1);
if (!ctx) {
    fprintf(stderr, "Failed to create the context: %s\n", modbus strerror(errno));
    return -1;
}
if (modbus connect(ctx) == -1) {
    fprintf(stderr, "Unable to connect: %s\n", modbus strerror(errno));
    modbus free(ctx);
    return -1;
}
//Set the Modbus address of the remote slave
modbus_set_slave(ctx, 2);
                             //modbus_set_slave(ctx, REMOTE_ID);
//Read 5 holding registers
num = modbus write bit(ctx, 2, 1); //0x5 ---> modbus_write_bit(modbus_t *ctx, int addr, int status);
modbus close(ctx);
modbus free(ctx);
```

步驟 4:於 LinPAC 上執行 Modbus 程式

指令:

# ./hello-rtu

Modbus Development kit for Linux

第11頁

# 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 (模組的設置參數可參考手冊第 7 頁),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。



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
```

Modbus Development kit for Linux

版本:1.0.2

第12頁

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 rtusiave 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/tty02'通訊埠	
der 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))	
· · · · · · · · · · · · · · · · · · ·	
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/exam	ples# python rtumaster_example.py
2018-12-10 17:47:25,575 INFO modbus	_rtuinit MainThread RtuMaster /dev/tty02 is opened
2018-12-10 17:47:25,578 INFO rtumas	ter_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 rtumas	ter_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(模組的設置參數可參考手冊第 7 頁),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 的相依模组。

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

步驟 3: 使用以下指令,安装 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。



**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
```

Modbus Development kit for Linux

版本:1.0.2

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(
          => '/dev/ttyO2', // 通訊埠
   port
                      // 每秒傳輸位元
   baudrate => 9600,
                      // 同位檢查
   parity => 'none',
);
   修改 write new addr.pl 的 Modbus 命令設定,並設置 AO 通道的值。
// Modbus 功能碼
my $req = $client->write single register(
                       // 從站設備的站號 (NetID)
   unit
          => 1,
                      // 通道位址
   address => 32,
        => 65535 // 設定值
   value
);
  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
                       // 從站設備的站號 (NetID)
   unit
          => 1,
                       // 通道位址
   address => 32.
                       // 所要操作的暫存器長度
   quantity => 1,
);
```

```
範例程式 write_new_addr.pl
```



**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-gnueabihf/bin:/c/cygwin/LinPAC\_AM335x\_SDK/Linaro\_GCC\_4.7/arm-linux-gnueabihf/lib c/usr/lib/opkg/alternatives:/c/cygwin/LinPAC\_AM335x\_SDK/Linaro\_GCC\_4.7/arm-linux-gnueabihf/libc/usr/li b/pkgconfig'

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



版本:1.0.2

## 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。

<u>M</u> ~	_		$\times$
Cindy@RD1-Freda-Chen MINGW32 ~ \$ export  grep \$PATH declare -x PATH="/mingw32/bin:/usr/local/bin:/usr/bin:/c/Windows/System32: indows/System32/Wbem:/c/Windows/System32/WindowsPowerShell/v1.0/:/usr/bin/site_ endor_perl:/usr/bin/core_perl:/c/cygwin/LinPAC_AM335x_SDK/Linaro_GCC_4.7/bin:/c AM335x_SDK/Linaro_GCC_4.7/arm-linux-gnueabihf/bin:/c/cygwin/LinPAC_AM335x_SDK/L rm-linux-gnueabihf/libc/usr/lib/opkg/alternatives:/c/cygwin/LinPAC_AM335x_SDK/L rm-linux-gnueabihf/libc/usr/lib/pkgconfig"	/c/Win _perl:/ /Cygwi _inaro_ _inaro_	ndows:/ /usr/bi in/LinF _GCC_4. _GCC_4.	/c/W in/v PAC_ .7/a .7/a
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/libn	nodbus-3.1.10	)	- 0	×
Cindy@RD1-Freda-Chen MINGW32 /c/ \$ pacman -S autotools resolving dependencies looking for conflicting packages	cygwin/LinP	AC_AM335x_SDK/lib	modbus-3.1.10	
Packages (10) automake-wrapper-1 automake1.13-1.13. automake1.16-1.16.	1-4 automa 4-7 automa 5-1 libltd	ke1.11-1.11.6-6 ke1.14-1.14.1-6 N-2.4.6-14 libto	automake1.12-1.12.6-6 automake1.15-1.15.1-4 ol-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 K1B	86.4 K1B/S 00:06	[#####################################	100%
automake1.12-1.12.6-6-any	503.1 K1B	/5.6 K1B/S 00:0/	<u> </u>	100%
automaker. $11-1.11.0-0-any$	490.2 K1B	341 K1B/S 00:01	[#####################################	100%
$1101101-2.4.0-14-X80_04$	32.0 K1B	23.5 K1B/S 00:01	[#####################################	100%
$1101001-2.4.0-14-x00_04$	500.1 KID	233 KIB/S 00:02	[#####################################	100%
$automake1 14_1 14 1_6_any$	502 1 Kip	60.5  kig/s 00.08		100%
automake_wrapper_11_1_apy	JUS.1 KID	1210 B/c 00.00	[#####################################	100%
automake-wiapper-11-4-any	4.4 KID	651 B/s 00:04	[#####################################	100%
Total $(10/10)$	2.J KIB 2 / MiR	261 KiB/s 00:04	[#####################################	100%
(10/10) checking keys in keyring	5.4 110	201 KIB/S 00.13		100%
(10/10) checking keys in keyring $(10/10)$ checking package integri	tv		[#####################################	100%
(10/10) loading package files	cy		<b>F###################################</b>	100%
(10/10) checking for file confli	cts		[######################################	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-wrap	per		[####################################	100%
( 8/10) installing libltdl			[######################################	100%
( 9/10) installing libtool			[######################################	100%
(10/10) installing autotools			[######################################	100%
:: Running post-transaction hooks	s			
(1/1) Updating the info directory	v file			

輸入 "gcc −v" 指令 · 檢查 gcc 版本。

l	<u>M</u> ~	_		×
ł	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/configureprefix=/mingw32with-local-prefix=/mingw32/local	bui	ld=i680	5-w6
	4-mingw32host=i686-w64-mingw32target=i686-w64-mingw32with-native-system-header-dir=/m	ringw32	2/inclu	ude
	libexecdir=/mingw32/libenable-bootstrapenable-checking=releasewith-arch=i686with	-tune=	=generi	ic -
	-enable-languages=c,lto,c++,fortran,ada,objc,obj-c++,jitenable-sharedenable-staticena	ble-1	ibatomi	ic -
	-enable-threads=posixenable-graphiteenable-tully-dynamic-stringenable-libstdcxx-tiles	ystem	-tse	enab
	le-libstdcxx-timedisable-libstdcxx-pchenable-libenable-libgompdisable-multilibc	isable	e-rpath	1
	disable-win32-registrydisable-nisdisable-werrordisable-symverswith-libiconvwith	-syste		) (
	with-gmp=/mingws2with-mprr=/mingws2with-mpc=/mingws2with-isi=/mingws2with-pkgversi	on= Ke	ev4, ві	111C 14
	by Marsz projectwith-buguri=https://github.com/msysz/Mindw-packages/issueswith-gu-as- disabla_libetdovy.dobugdisabla_sili_excentionswith_dwamf2with_boot_ldflags_static_	ibstd	n-gnu-	na -
	-unsaule-fibstuc-static-libstuc-s	IDSCU		VI CII
	Thread model: nosi			
	Supported ITO compression algorithms: zlib zstd			
	acc version 12.2.0 (Rev4, Built by MSYS2 project)			
	Cindy@RD1-Freda-Chen MINGW32 ~			
	\$			
1				

M ~	_		×
Cindy@RD1-Freda-Chen MINGW32 ~			
\$ arm-linux-gnueabihf-gcc -v			
specs			
CULLECI_GUC=C: Cygwin(LinPAC_AM335x_SUK(Linaro_GUC_4./(Din(arm-innux-gnueabint-gcc.exe	ibf /A	7 3/1	to-w
ranger_exe	1111 / 4	. / . 3/ 1	CO-w
arm-linux-gnueabihf			
/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabihf-win32/.build/src/gcc-linaro-4.7-2013.	03/co	nfigur	'e
build=i686-build_pc-linux-gnuhost=i586-host_pc-mingw32msvctarget=arm-linux-gnueabihfprefi	x=/cb	uild/s	lave
s/oorts/crosstool-ng/builds/arm-linux-gnueabiht-win32/installwith-sysroot=/cbuild/slaves/oorts/	cross	tool-n	ig/bu
1105/arm=11nux=gnueab1nr=w1n52/1nstall/arm=11nux=gnueab1nr/11bc==enable=languages=c,c++,tortran =	-enab	re-mui	01_N
G linaro-1.13.1-4.7-2013.03-20130313 - Linaro GCC 2013.03'with-bugurl=https://bugs.launchpad.ne	t/acc	-linar	0
enablecxa_atexitenable-libmudflapenable-libgompenable-libsspwith-gmp=/cbuild/slaves,	/oort:	s/cros	stoo
l-ng/builds/arm-linux-gnueabihf-win32/.build/arm-linux-gnueabihf/build/staticwith-mpfr=/cbuild/	slave	s/oort	s/cr
osstool-ng/builds/arm-linux-gnueabihf-win32/.build/arm-linux-gnueabihf/build/staticwith-mpc=/cb	uild/s	slaves	/oor
ts/crosstool-ng/builds/arm-linux-gnueabint-win32/.build/arm-linux-gnueabint/build/staticwith-pp	l=/cbi	u11d/s	lave buil
d/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabihf-win32/.build/arm-linux-gnueabihf/build/stat	ic	with-1	ibel
f=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabihf-win32/.build/arm-linux-gnueabihf/bu	ild/s	tatic	wi
th-host-libstdcxx='-L/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabihf-win32/.build/arm	-linu	x-gnue	abih
f/build/static/lib -lpwl'enable-threads=posixdisable-libstdcxx-pchenable-linker-build-id	enal	ble-go	ld -
-with-local-prefix=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabiht-win32/install/arm-	l ı nux·	-gnuea	ıb1h†
/ TDCenable-C99enable-long-longwith-mode=thumb			
gcc version 4.7.3 20130226 (prerelease) (crosstool-NG linaro-1.13.1-4.7-2013.03-20130313 - Linaro	GCC 20	013.03	)
Cindy@RD1-Freda-Chen MINGW32 ~			

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

# 編譯 LIBMODBUS 原始碼及其範例

## 首先執行"sh autogen.sh"以產生設定腳本。

/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/ltversion.m4' libtoolize: linking file 'm4/ltversion.m4' libtoolize: linking file 'm4/ltversion.m4' libtoolize: linking file 'm4/ltvobsolete.m4' configure.ac:33: installing 'build-aux/compile' configure.ac:56: installing 'build-aux/config.guess' configure.ac:32: installing 'build-aux/install-sh' configure.ac:32: 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:



#### 使用"make"指令來編譯程式。

M /c/cygwi	n/LinPAC_AM335x_SDK/libmodbus-3.1.10	(24)		×
Cindy@MD1-4 \$ make makeno-4 Making all CC	reda-Chen MINGW32 /c/cygwin/LinPAC_AM335x_ print-directory all-recursive in src modbus.lo modbus.data_lo	SDK/libmodbus-3	.1.10	
CC	modbus-rtu.lo			
CCLB	modbus-tcp.lo libmodbus la			
Making all	in tests			
CC CC	bandwidth-server-one.o			
CCLD	bandwidth-server-one			
CC	bandwidth-server-many-up.o			
CC	bandwidth-client.o			
CCLD	bandwidth-client			
CC	random-test-server.o			
CCLD	random-test-server			
CC	random-test-client.o			
CCLD	random-test-citent			
CCLD	unit_test_server			
CC	unit-test-client o			
CCLD	unit-test-client			
CC CCLD	version.o version			

# 使用"make install"命令安裝 Modbus 函式庫及其表頭檔:

//////////////////////////////////////	-	_	×
Cindy@RDi-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/libtoolmode=install /usr/bin/install -c libmodbus.la '/c/cygwin/LinPAC_AM335x_SDK/lil	anodbu	5-3.1	.10/1
<pre>inpac/lib' libtool: install: /usr/bin/install ~c .libs/libmodbus.so.5.1.0 /c/cygwin/LinPAC_AM335x_SDK/libmodbus-3.:</pre>	1.10/1	inpac	/1ib/
<pre>Timoodbus.so.5.1.0 libtool: install: (cd /c/cygwin/LinPAC_AM335x_SDK/libmodbus.3.1.10/linpac/lib &amp;&amp; { cp -pR -f libmodbus.s us.so.5    { rm -f libmodbus.so.5 &amp;&amp; cp -pR libmodbus.so.5.1.0 libmodbus.so.5; }; }) libtool: install: (cd /c/cygwin/LinPAC_AM335x_SDK/libmodbus.3.1.10/linpac/lib &amp;&amp; { cp -pR -f libmodbus.so us.so    { rm -f libmodbus.so &amp;&amp; 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/ dbus.la</pre>	so.5.1 so.5.1 linpac	.0 li .0 li /lib/	bmodb bmodb libmo
<pre>libtool: install: /usr/bin/install -c .libs/libmodbus.a /c/cygwin/LinPAC_AM335x_SOK/libmodbus-3.1.10/lin us.a</pre>	npac/1	ib/li	bmodb
<pre>Usia Tibtool: install: chmod 644 /c/cygwin/LinPAC_AM335x_SDK/Tibmodbus-3.1.10/Tinpac/Tib/Tibmodbus.a Tibtool: install: arm-Tinux-gnueabihf-ranTib /c/cygwin/LinPAC_AM335x_SDK/Tibmodbus-3.1.10/Tinpac/Tib/Tib Tibtool: finish: PATH="/mingw32/bin:/usr/Tocal/bin:/usr/bin:/bin:/c/windows/System32:/c/Windows/c/windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/System32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:/c/Windows/cystem32:</pre>	wodbu ws/Sy 1:/c/c cygwin inaro 10/lin	s.a stem3 ygwin /LinP _GCC_ pac/l	2/wbe /LinP AC_AM 4.7/a ib
Libraries have been installed in: /c/cygwin/LinPAC_AM335x_SDK/libmodbus-3.1.10/linpac/lib			
<pre>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:</pre>			
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_AM335 3.1.10/linpac/include/modbus' Making install in tests	_SDK/	libmo	dbus-
<pre>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,</pre>	/share	/doc/	libmo
/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/pkgcon	ıfig'		

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

modbus		3   lib	
C\cygwin\LinPAC_AM335x_S	DK\libmodbus-3.1.8\linpac\include\modbu	C\cygwin\LinPAC_AM335x_SE	K\libmodbus-3.1.8\linpac\lib
File name	Size	File name	Size
modbus.h	12 KB	[] libmodbus.a	151 KB
modbus-rtu.h	2 KB	[] libmodbusJa	1 KB
modbus-tcp.h	2 KB	ibmodbus.so	125 KB
modbus-version.h	3 KB	libmodbus.so.5	125 KB
		libmodbus.so.5.1.0	125 KB
		pkgconfig	

Cindy@RD1-Freda-Chen MINGW32 /c/cygwin/LinPAC\_AM335x\_SDK/libmodbus-3.1.8/tests \$ arm-linux-gnueabihf-gcc -I../linpac/include/modbus -Im -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 = -Im
- 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
```



P COM1 - PuTTY —	$\times$
icpdas login: root	
Password:	
Last login: Wed Dec 28 08:51:54 UTC 2022 on tty05	
Welcome to Ubuntu 12.04.4 LTS (GNU/Linux 3.2.14-rt24 armv71)	
* Documentation: https://help.ubuntu.com/	
root@icpdas:~#	
root@icpdas:~# file version.exe	
version.exe: ELF 32-bit LSB executable, ARM, version 1 (SYSV), dynamically lin	ke
d (uses shared libs), for GNU/Linux 2.6.32, BuildID[sha1]=0xf77e2fe8bb34a31551	19
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?
V1.0.2	October 2024	新增libmodbus於Linux PC之操作應用