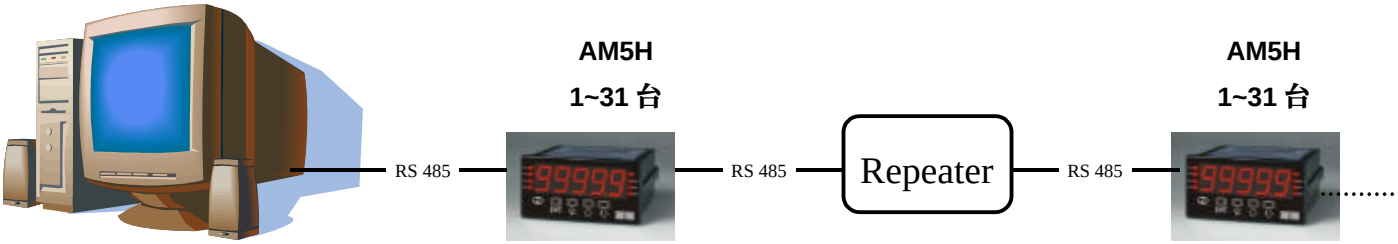
ModBus RTU Mode <<通信協定簡介>>

一簡介

AM5H 系列 採用 MODBUS 通訊協定，一台電腦可連接 1~32 台電表，或經由訊號擴大器（Repeater）可延伸至 255 台電表（圖一）。

MODBUS 使用兩種傳送模式，一種為 ASCII Mode，一種為 RTU Mode，AM5H 系列使用 RTU Mode，通訊方式為半雙工方式（Half-Duplex）。



（圖一）

AM5H 系列使用的通訊介面：RS-485。

二 MODBUS RTU MODE 簡介

1.基本命令結構（均為 16 進制 Hexadecimal）

START OF FRAME	ADDRESS FIELD	FUNCTION CODE	DATA FIELD	ERROR CHECK	END OF FRAME

（1）START OF FRAME：至少有 4 個字元的時間沒有傳送資料。

（2）ADDRESS FILED：欲讀取或控制 Meter 的位址（位址範圍為 1~255），Address 0 為廣播方式，只對 Function Code：06H 有效。

（3）FUNCTION CODE：

a. 03H：讀取 Meter 的資料。

b. 06H：將資料寫入 Meter。

（4）DATA FIELD：包括暫存器位址及欲讀取之 Word 數。

（5）ERROR CHECK：16bit CRC，計算方式詳述於後面章節。

（6）END OF FRAME：至少有 4 個字元的時間沒有傳送資料。

2.Bit Per Byte

分為下列 3 種

Start Bit	Data Bit	Parity	Stop
1	8	None	2
1	8	Even	1
1	8	Odd	1

三 CRC 計算方式

CRC 計算方式有兩種，一種為邏輯運算，另一種為查表方式，AM5H 系列採用邏輯運算方式。CRC 欄位為 2 個 16 進制 Byte，從 ADDRESS FIELD 計算至 DATA FIELD 結束，若 PC 計算之 CRC 與接收的不符，則表示資料錯誤。

1. 邏輯運算

   計算步驟如下：

   （1）將一個 16 位元暫存器填入 FFFF（Hex），我們定義為 CRC 暫存器。

   （2）將 CRC 暫存器的低 8 位元與 Message 的第一個 Byte 做互斥或（Exclusive OR），結果放入 CRC 暫存器。

   （3）將 CRC 暫存器向右移一個位元，CRC 暫存器最高位元填入 0，比較移出的位元定義為 SLSB ）。

   （4）若 SLSB=0，重覆步驟 3。若 SLSB=1，將 CRC 暫存器與常數 A001（Hex）做互斥或，結果放入 CRC 暫存器。

   （5）重覆步驟 3 及步驟 4，直到 8 位元都做完。

   （6）重覆步驟 2~5 ，直到所有 Byte 都做完。

   （7）計算出來 CRC 的值需高低位元互換填入 Message 中。

Addr	Func	Data Count	Data	Data	Data	Data	CRC Lo	CRC Hi

2. 查表方式

```
/* The function returns the CRC as a type unsigned short. */
/* CRC Generation Function */
unsigned short CRC16 (puchMSG, usDataLen)
unsigned char *puchMsg：                    /* message to calculate CRC upon */
unsigned short usDataLen：                  /* quantity of bytes in message */

 {

   unsigned char uchCRCHi = 0xFF;           /* high CRC byte initialized */
   unsigned char uchCRCLo = 0xFF;           /* low CRC byte initialized */
   unsigned uIndex;                         /* will index index into CRC lookup */


   while (usDataLen--)                      /* pass through message buffer */

 {

uIndex=uchCRCHi ^ *puchMsgg++；             */ calculate the CRC */
uchCRCHi = uchCRCLo ^auchCRCHi〔uIndex〕；
uchCRCLo = auchCRCLo〔uIndex〕；

 }

return（uchCRCHi<< 8 | uchCRCLo）；

 }
```

High Order Byte Table

```
/* Table of CRC values for high - order byte */
static unsigned char auchCRCHi [] = {
 0x00 , 0xC1 , 0x81 , 0x40 , 0x01 , 0xC0 , 0x80 , 0x41 , 0x01 , 0xC0 ,
 0x80 , 0x41 , 0x00 , 0xC1 , 0x81 , 0x40 , 0x01 , 0xC0 , 0x80 , 0x41 ,
 0x00 , 0xC1 , 0x81 , 0x40 , 0x00 , 0xC1 , 0x81 , 0x40 , 0x01 , 0xC0 ,
 0x80 , 0x41 , 0x01 , 0xC0 , 0x80 , 0x41 , 0x00 , 0xC1 , 0x81 , 0x40 ,
 0x00 , 0xC1 , 0x81 , 0x40 , 0x01 , 0xC0 , 0x80 , 0x41 , 0x00 , 0xC1 ,
 0x81 , 0x40 , 0x01 , 0xC0 , 0x80 , 0x41 , 0x01 , 0xC0 , 0x80 , 0x41 ,
 0x00 , 0xC1 , 0x81 , 0x40 , 0x01 , 0xC0 , 0x80 , 0x41 , 0x00 , 0xC1 ,
 0x81 , 0x40 , 0x00 , 0xC1 , 0x81 , 0x40 , 0x01 , 0xC0 , 0x80 , 0x41 ,
 0x00 , 0xC1 , 0x81 , 0x40 , 0x01 , 0xC0 , 0x80 , 0x41 , 0x01 , 0xC0 ,
 0x80 , 0x41 , 0x00 , 0xC1 , 0x81 , 0x40 , 0x00 , 0xC1 , 0x81 , 0x40 ,
 0x01 , 0xC0 , 0x80 , 0x41 , 0x01 , 0xC0 , 0x80 , 0x41 , 0x00 , 0xC1 ,
 0x81 , 0x40 , 0x01 , 0xC0 , 0x80 , 0x41 , 0x00 , 0xC1 , 0x81 , 0x40 ,
 0x00 , 0xC1 , 0x81 , 0x40 , 0x01 , 0xC0 , 0x80 , 0x41 , 0x01 , 0xC0 ,
 0x80 , 0x41 , 0x00 , 0xC1 , 0x81 , 0x40 , 0x00 , 0xC1 , 0x81 , 0x40 ,
 0x01 , 0xC0 , 0x80 , 0x41 , 0x00 , 0xC1 , 0x81 , 0x40 , 0x01 , 0xC0 ,
 0x80 , 0x41 , 0x01 , 0xC0 , 0x80 , 0x41 , 0x00 , 0xC1 , 0x81 , 0x40 ,
 0x00 , 0xC1 , 0x81 , 0x40 , 0x01 , 0xC0 , 0x80 , 0x41 , 0x01 , 0xC0 ,
 0x80 , 0x41 , 0x00 , 0xC1 , 0x81 , 0x40 , 0x01 , 0xC0 , 0x80 , 0x41 ,
 0x00 , 0xC1 , 0x81 , 0x40 , 0x00 , 0xC1 , 0x81 , 0x40 , 0x01 , 0xC0 ,
 0x80 , 0x41 , 0x00 , 0xC1 , 0x81 , 0x40 , 0x01 , 0xC0 , 0x80 , 0x41 ,
 0x01 , 0xC0 , 0x80 , 0x41 , 0x00 , 0xC1 , 0x81 , 0x40 , 0x01 , 0xC0 ,
 0x80 , 0x41 , 0x00 , 0xC1 , 0x81 , 0x40 , 0x00 , 0xC1 , 0x81 , 0x40 ,
 0x01 , 0xC0 , 0x80 , 0x41 , 0x01 , 0xC0 , 0x80 , 0x41 , 0x00 , 0xC1 ,
 0x81 , 0x40 , 0x00 , 0xC1 , 0x81 , 0x40 , 0x01 , 0xC0 , 0x80 , 0x41 ,
 0x00 , 0xC1 , 0x81 , 0x40 , 0x01 , 0xC0 , 0x80 , 0x41 , 0x01 , 0xC0 ,
 0x80 , 0x41 , 0x00 , 0xC1 , 0x81 , 0x40
} ;
```

Low Order Byte Table

```
/* Table of CRC values for low - order byte */
static char auchCRCLo[] = {
 0x00 , 0xC0 , 0xC1 , 0x01 , 0xC3 , 0x03 , 0x02 , 0xC2 , 0xC6 , 0x06 ,
 0x07 , 0xC7 , 0x05 , 0xC5 , 0xC4 , 0x04 , 0xCC , 0x0C , 0x0D , 0xCD ,
 0x0F , 0xCF , 0xCE , 0x0E , 0x0A , 0xCA , 0xCB , 0x0B , 0xC9 , 0x09 ,
 0x08 , 0xC8 , 0xD8 , 0x18 , 0x19 , 0xD9 , 0x1B , 0xDB , 0xDA , 0x1A ,
 0x1E , 0xDE , 0xDF , 0x1F , 0xDD , 0x1D , 0x1C , 0xDC , 0x14 , 0xD4 ,
 0xD5 , 0x15 , 0xD7 , 0x17 , 0x16 , 0xD6 , 0xD2 , 0x12 , 0x13 , 0xD3 ,
 0x11 , 0xD1 , 0xD0 , 0x10 , 0xF0 , 0x30 , 0x31 , 0xF1 , 0x33 , 0xF3 ,
 0xF2 , 0x32 , 0x36 , 0xF6 , 0xF7 , 0x37 , 0xF5 , 0x35 , 0x34 , 0xF4 ,
 0x3C , 0xFC , 0xFD , 0x3D , 0xFF , 0x3F , 0x3E , 0xFE , 0xFA , 0x3A ,
 0x3B , 0xFB , 0x39 , 0xF9 , 0xF8 , 0x38 , 0x28 , 0xE8 , 0xE9 , 0x29 ,
 0xEB , 0x2B , 0x2A , 0xEA , 0xEE , 0x2E , 0x2F , 0xEF , 0x2D , 0xED ,
 0xEC , 0x2C , 0xE4 , 0x24 , 0x25 , 0xE5 , 0x27 , 0xE7 , 0xE6 , 0x26 ,
 0x22 , 0xE2 , 0xE3 , 0x23 , 0xE1 , 0x21 , 0x20 , 0xE0 , 0xA0 , 0x60 ,
 0x61 , 0xA1 , 0x63 , 0xA3 , 0xA2 , 0x62 , 0x66 , 0xA6 , 0xA7 , 0x67 ,
 0xA5 , 0x65 , 0x64 , 0xA4 , 0x6C , 0xAC , 0xAD , 0x6D , 0xAF , 0x6F ,
 0x6E , 0xAE , 0xAA , 0x6A , 0x6B , 0xAB , 0x69 , 0xA9 , 0xA8 , 0x68 ,
 0x78 , 0xB8 , 0xB9 , 0x79 , 0xBB , 0x7B , 0x7A , 0xBA , 0xBE , 0x7E ,
 0x7F , 0xBF , 0x7D , 0xBD , 0xBC , 0x7C , 0xB4 , 0x74 , 0x75 , 0xB5 ,
 0x77 , 0xB7 , 0xB6 , 0x76 , 0x72 , 0xB2 , 0xB3 , 0x73 , 0xB1 , 0x71 ,
 0x70 , 0xB0 , 0x50 , 0x90 , 0x91 , 0x51 , 0x93 , 0x53 , 0x52 , 0x92 ,
 0x96 , 0x56 , 0x57 , 0x97 , 0x55 , 0x95 , 0x94 , 0x54 , 0x9C , 0x5C ,
 0x5D , 0x9D , 0x5F , 0x9F , 0x9E , 0x5E , 0x5A , 0x9A , 0x9B , 0x5B ,
 0x99 , 0x59 , 0x58 , 0x98 , 0x88 , 0x48 , 0x49 , 0x89 , 0x4B , 0x8B ,
 0x8A , 0x4A , 0x4E , 0x8E , 0x8F , 0x4F , 0x8D , 0x4D , 0x4C , 0x8C ,
 0x44 , 0x84 , 0x85 , 0x45 , 0x87 , 0x47 , 0a46 , 0x86 , 0x82 , 0x42 ,
 0x43 , 0x83 , 0x41 , 0x81 , 0x80 , 0x40
};
```

註：CRC 在某些特定位址，邏輯運算與查表方式會有所不同，請特別注意。