

// 下面二種方法, 處理 Modbus CRC-16 演算

/\* 方法 1: Modbus CRC16 計算法 \*/

//CRC-16:  $x^{16} + x^{15} + x^2 + 1$

檢查碼由 Address 到 Data content 結束。其運算規則如下：

步驟 1：令 16-bit 寄存器 (CRC 暫存器) = FFFFH.

步驟 2：Exclusive OR 第一個 8-bit byte 的訊息指令與低位元 16-bit CRC 寄存器, 做 Exclusive OR，將結果存入 CRC 寄存器內。

步驟 3：又移一位 CRC 寄存器，將 0 填入高位處。

步驟 4：檢查右移的值，如果是 0, 將步驟 3 的新值存入 CRC 寄存器內, 否則 Exclusive OR A001H 與 CRC 寄存器，將結果存入 CRC 寄存器內。

步驟 5：重複步驟 3~步驟 4，將 8-bit 全部運算完成。

步驟 6：重複步驟 2~步驟 5，取下一個 8-bit 的訊息指令，直到所有訊息指令運算完成。

最後，得到的 CRC 寄存器的值，即是 CRC 的檢查碼。值得注意的是 CRC 的檢查碼必須交換放置於訊息指令的檢查碼中。

\*/

// =====

Unsigned int crc\_chk(unsigned char\* data, unsigned char length)

{

int j;

unsigned int reg\_crc=0xFFFF;

while(length--)

{

reg\_crc ^= \*data++;

for(j=0;j<8;j++)

{

if(reg\_crc & 0x01) /\* LSB(b0)=1 \*/

reg\_crc=(reg\_crc>>1) ^ 0xA001;

else

reg\_crc=reg\_crc >>1;

}

}

return reg\_crc;

}

// =====

// =====

// =====

```

/* 方法 2: Modbus CRC16 查表法 */
// =====
/* CRC16 Generation Function with 'C' language*/
/* Msg:*message to calculate CRC upon*/
/* usDatalen: number of bytes in message*/
unsigned int CRC16(char *Msg, unsigned char usDatalen)
{
    unsigned char uchCRCHi=0xFF ; /*CRC high byte*/
    unsigned char uchCRCLo=0xFF ; /*CRC low byte*/
    unsigned char uIndex;
    while(usDatalen--)          /*pass through message buffer*/
    {
        uIndex=uchCRCHi^*Msg++; /*calculate the CRC*/
        uchCRCHi=uchCRCLo^auchCRCHi[uIndex] ;
        uchCRCLo=auchCRCLo[uIndex] ;
    }
    return (uchCRCHi<<8 | uchCRCLo) ;
}

```

```

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,
x80,0x41,0x00,0xc1,0x81,0x40,0x00,0xc1,0x81,0x40,
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,
0x80,0x41,0x01,0xc0,0x80,0x41,0x00,0xc1,0x81,0x40,
0x00,0xc1,0x81,0x40,0x01,0xc0,0x80,0x41,0x01,0xc0,
0x80,0x41,0x01,0xc0,0x80,0x41,0x00,0xc1,0x81,0x40,
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};
```

```
static unsigned 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,  
0x44,0x84,0x85,0x45,0x87,0x47,0x46,0x86,0x82,0x42,  
0x43,0x83,0x41,0x81,0x80,0x40};
```