

分類/Classification	<input type="checkbox"/> tDS	<input type="checkbox"/> tGW	<input type="checkbox"/> PETL/tET/tPET	<input type="checkbox"/> DS/PDS/PPDS	<input checked="" type="checkbox"/> TouchPAD
作者/Author			日期/Date		編號/NO. FAQ005
	<input type="checkbox"/> I/O Card	<input type="checkbox"/> VXC Card	<input type="checkbox"/> VxComm	<input type="checkbox"/> Other	

## Q: How to convert a Modbus AI or AO Data (Register) into a value we want?

A: The Modbus is a protocol which leaves the data format undefined. Users can/should define their own. The followings we take two examples to illustrate two common situations:

### 1. Linear Mapping:

Supposed that we have an AO output ranging from -10 ~ 10V, the Modbus uses two bytes to represent this data and then we have:

-10      v      10  
-32768    reading    32767 (0x7FFF)

Through linear mapping, we have the remote side AO output v:

,where reading is an unsigned 16 bit integer received from a Modbus channel.

That is,

### 2. Byte Swap for Floating-point number:

Using the Modbus protocol, the most significant byte (MSB) is transmitted first in a word (two bytes) but the float data type has four bytes with little-endian format (where LSB is stored first) in Intels CPU. Then we have the followings:

Supposed

In the first transmit through Modbus: bytes[0], bytes[1]

In the second transmit though Modbus: bytes[2], bytes[3]

Then the byte order for Intels CPU is

bytes[1], bytes[0], bytes[3], bytes[2]

Users can now easily put the four bytes together to form a float value.

Code snippet in C as below:

```
int ivalue = (bytes[2]<<24)+( bytes[3]<<16)+( bytes[0]<<8)+ bytes[1];  
float fvalue = *(float*)&ivalue;
```

*P.S. Different platforms (PC/PLC) may have different data format.*