字节与二进制

计算机底层运行都是二进制,最小单位是位

对于开发者来说,我们操作的最小单位是字节

	位/Bool	1	True/False
	字节/Byte	8	0-255
	字节/SByte	8	-128-127
	短整型/UShort	16	0-65535
	短整型/Short	16	-32768-32767
	整型/UInt	32	0-4294967295
	整型/Int	32	2147483648~2147483647
	长整型/ULong	64	0-2^64
	长整型/Long	64	-2^63-2^63-1
	单精度/Float	32	-3.4E38 ~ 3.4E38
	双精度/Double	64	-1.79E308 ~ +1.79E308
	字节串/String		 无



1.2 数据结构原理及大小端 の

大端字节顺序是指高位字节存储在低位地址,低位字节存储在高位地址
MSB 低 Byte 地址 LSB 高

Ox11 0x22 0x33 0x44 0x88 0x99 0xab 0xcd

LIII 122 33 44) short(88 99) short(ab cd)

小端字节顺序则反之,高位字节存储在高位地址,低位字节存储在低位地址
LSB 低 Byte 地址 MSB 高

Ox44 0x33 0x22 0x11 0x99 0x88 0xcd 0xab

LIII 122 33 44) short(88 99) short(ab cd)

为什么会存在大小端

"服务器"按照它的大小端方式把数据转换成字节数组发给我们

数据转换为什么需要大小端

如何使用C#代码来举例说明

我们必须要按照"服务器"的大小端方式来解析数据,才能得到正确的数据

大小端的本质就是一个字节顺序

BitConvet.GetBytes

BitConvet.ToSingle

ABCD

大小端几种形式

大小端字节序

CDAR

BADC

CDAB

DCBA

大小端只是一种字节顺序,对于一个已知 的"服务器"来说,是一个固定值

怎么理解大小端

我们只需要使用浮点数或整数,做一个测试,即可快速确认



助教微信号: xiketang777 微信公众号: dotNet工控上位机