

# 详解STEP7的指针数据类型(POINTER)

原创文章，转载请注明出处。

更多实用资料请登录方正智芯官网：[www.founderchip.com](http://www.founderchip.com)

作者：北岛李工

西门子STEP7（博途）编程开发环境给了程序设计人员很大的自由度，丰富的指令和数据类型让工程师们可以尽情的发挥自己的聪明才智，编写出高效、稳定、功能强大的代码。今天和大家聊聊STEP7编程的比较深入的知识——指针（POINTER）数据类型。



指针的概念最早出现于C语言，表示一个变量的地址。指针变量占用四个字节（32位操作系统），可以让程序员很方便的对内存进行读写，大大提高了程序的效率。C语言也因为支持指针，而被用来编写操作系统。大名鼎鼎的Linux操作系统，就是用C语言编写的。

在PLC的世界里，西门子的STEP7（博途）也引入了指针的概念。不过STEP7中的指针类型和C语言的有所不同，它占用了6个字节，如下图：

Byte 0	DB Number or 0												Byte 1			
Byte 2	Memory Area												Byte 3			
Byte 4	B	B	B	B	B	B	B	B	B	B	B	B	X	X	X	Byte 5

其中，Byte0和Byte1用来表示数据块（DB）的编号，如果指向的存储区不是DB块，则值为0；Byte2是要访问的存储区的代码（Memory Area Code）；Byte3的低3位、Byte4和Byte5的高5位用来表示变量的字节地址（图中蓝色部分）；Byte5的低3位表示变量的位的地址（图中棕色部分）；可见，指针类型能清晰的表达要指向的数据的区域（Memory Area）、字节（Byte address）及位（bit）的信息。

关于存储区（Memory Area）代码，请看下面这张表格：

方正智芯(founder chip) - STEP7指针类型存储区代码表

Hexadecimal Code	Memory Area	Description
B#16#80	P	S7-300/400外设(Peripherals)
B#16#1	P	S7-1500外设输入(Peripherals Input)
B#16#2	P	S7-1500外设输出(Peripherals output)
B#16#81	I	输入存储区 (input)
B#16#82	Q	输出存储区 (output)
B#16#83	M	位存储区 (bit memory)
B#16#84	DBX	全局数据存储区 (DB)
B#16#85	DIX	背景数据存储区 (DI)
B#16#86	L	局部数据存储区 (L)
B#16#87	V	先前局部数据存储区 (Previous L)

左边一栏是存储区的十六进制代码，中间一栏是存储区的名称，右边一栏是相应的描述。需要注意的是，S7-300/400的外设存储区 (P) 代码为“B#16#80”，而在S7-1500中，对外设输入和外设输出进行了区分，“B#16#1”表示输入 (PI)，“B#16#2”表示输出 (PQ)；

在STEP7中，指针变量用符号“P#”标识，比如“P#DB100.DBX1.0”，表示一个指向DB100的第1个字节的第0位的指针变量。STEP7支持四种类型的指针变量：

- 1) 存储区内指针 (Area-internal Pointer)：同一个存储区内的指针，比如：P#10.0；
- 2) 存储区间指针 (Cross-area Pointer)：指向存储区变量的指针，比如：P#M6.0。这里的存储区可以是输入存储区 (I)，输出存储区 (Q) 或位存储区 (M)；
- 3) 数据块指针 (DB Pointer)：指向数据块变量的指针，比如：DB101.DBX2.0；
- 4) 零指针 (Zero Pointer)：用来指向一个目前没有使用，将来可能用到的变量；

如果某个功能块 (FB或FC) 的形参是指针类型，那么当给形参赋值时，符号“P#”可以省略 (不爱偷懒的你也可以不省略哦)，STEP7会自动将输入的值转换成指针类型。

看完这篇文章，对指针是不是有较深的理解了呢？如果你想收藏本文，记得官网提供本文PDF版本下载哦。

相关参考文章：

[PLC基础篇之复杂数据类型：数组与字符串](#)

[PLC基础篇之数据类型 \(Data type\)](#)


方正智芯  
Founder Chip

长按扫码关注



方正智芯

公众号：founderchip

官方网站：www.founderchip.com

原创工业智能控制领域 (PLC、单片机、通信) 的技术分享