如果 FB 的传递参数（IN\_OUT）是一个复杂数据类型的变量，则它实际上是作为一个 6 字节的指针（48位）保存在背景数据块中的。这意味着它不再是一个变量，而是一个 6 字节的指针。由于指针指向了实参，因此它是必须被参数化。STEP 7 可以识别下面的复杂数据类型：“Date\_and\_Time”、“String”、“ Array”、“Struct”和“UDT”。

图 01 中提供了如何间接访问复杂数据类型的第一个字节的例程。在例子中介绍的程序可用作数据备份。


图 01

通过 IN\_OUT 变量 “Symbol\_des\_InOut” 装载 6 字节指针。数据块的块号由这个6 字节指针传送到临时变量“DB\_Num”中。通过语句“AUF DB[#DB\_Num]”，打开被保存的块号的数据块，然后将该地址装载到地址寄存器 AR1 中。

“+AR1”指令作用为将ACCU1中INT类型的值扩展为有符号24位数并与地址寄存器的内容相加。如果字节地址的最大范围超限 (0 ... 65535)也不会有影响，因为高位被剔除。

语句“L B[AR1,P#0.0]”中提供了被保存的数据和正确的地址。在例程中如果用语句“L Symbol\_des\_InOut[1]”代替， 那么数据直接来自于 IN\_OUT 参数，而不能从地址寄存器中装载被保存的数据， 即使这些数据的值可能已经发生了改变。请参考 STEP 7 在线帮助的相关信息。

* “Avoiding errors when calling blocks”。

**注意**例程的描述仅对 DB 区有效，因此用于其它区域 CPU 将进入停机状态。编号 0（DB0 )是不允许使用的。

“InOutvar.zip"”文件里包含了上面提到的例子程序，在一个 STEP 7 项目 下有两个 S7 站：

* 站：“ Beispiel in deutsch” 具有德文注释
* 站：“ Example in English” 具有英文注释

[InOutvar.zip](https://support.industry.siemens.com/cs/attachments/19106712/InOutvar.zip) ( 222 KB )

**给带有复杂数据类型的功能分配参数**复杂数据类型参数提供了一种在调用块和被调用块间传送大量相关数据的有效方式。一个区域或者一个结构可以作为复杂数据类型传送到被调用的功能中。

复杂数据类型变量只能保存在数据块或者本地数据堆栈中。因此，被调用块的实际参数必须定义在数据块(共享或者背景数据块)或者在本地数据堆栈中。

为了传送的目的，必须定义一个传送的实参，参数的类型与被调用的块类型一致。提供的这些参数(数据类型： ARRAY, STRUCT, DATE\_AND\_TIME  和 STRING) 必须使用符号方式。

LAD/STL/FBD 编辑器检查实际的参数与块的定义参数的兼容性。实际参数的传送是通过带有 DB 块号和地址的指针实现的。该指针通过 CALL 指令调用块时在临时堆栈生成。


图. 02

复杂数据类型必须仅保存在临时数据堆栈(FC12 的临时变量)或者在数据块 (DB5)中。