

安徽省滁州市SIEMENS西门子（授权）中国华东地区一级总代理商

产品名称	安徽省滁州市SIEMENS西门子（授权）中国华东地区一级总代理商
公司名称	广东湘恒智能科技有限公司
价格	.00/件
规格参数	变频器:西门子代理商 触摸屏:西门子一级代理 伺服电机:西门子一级总代理
公司地址	惠州大亚湾澳头石化大道中480号太东天地花园2栋二单元9层01号房（仅限办公）（注册地址）
联系电话	18126392341 15267534595

产品详情

该功能将多个字节（从“开始”开始，到“结束”结束）复制到一个字符串中。另外还将写入数据的长度复制到字符串标头中。

可以将任意类型的输入数据（数组、字符串等）附加到“DATA_IN”。必须在“DATA_OUT”上对字符串提前进行参数化。

功能说明：

数据块接受任意输入变量（数组、字符串或数据区域）并从输入区域DATA_IN的字节“开始”到“结束”将数据复制到输出区域“DATA_OUT”。“DATA_OUT”处的数据按字符串处理。

如果要从输入端的char数组中复制字节0到字节10（包括字节10），则必须指定数值“开始”=0；“结束”=10。0将数据复制到DATA_OUT的字符串中。

如果要将输入端的一个字符串从第一个字母复制到第12个字母则必须对“开始”=2和“结束”=13进行参数化。

如果需要复制的数据区域大于输出端上的字符串，则不会进行复制并将输出端PaFe设置到1。

如果DATA_OUT未进行参数化，则输出PaFe=2。PaFe=3，则输入数据过短（检查“开始”和“结束”，输入端“结束”上的数值可能位于输入端“开始”的数值前）。

然后可以使用其他系统功能（例如CONCAT）将输出的字符串附加到另一个字符串。或者也可以继续调整INSERT。

代码：

```

REGION (* Programm *) //Zeiger kopieren #DATA_IN_T := #DATA_IN; #DATA_OUT_T :=
#DATA_OUT; #dBZ_Hilfsvariable := #p_DATA_OUT.BZ;//BZ sichern //Zeiger anpassen um den
Header zu lesen/schreiben #p_DATA_OUT.DataCount := 1; //nur 1 Byte schreiben
//Laenge des Ausgabestring lesen #erg_SFC20 := BLKMOV(SRCBLK := #DATA_OUT_T, DSTBLK =>
#bHeader_Str_Laenge); #bHeader_Str_Inhalt := INT_TO_BYTE(#Ende - #Start + 1); #PaFE := 0;
IF BYTE_TO_INT(#bHeader_Str_Laenge) < (#Ende - #Start + 1) THEN #PaFE := 1;
//Ausgabedaten zu lang fuer Ausgabestring ELSIF #p_DATA_OUT.DBNR = 0 THEN #PaFE := 2;
//kein Ausgangfeld festgelegt ELSIF (#Ende - #Start + 1) < 1 THEN #PaFE := 3;
//Eingabedaten zu kurz ELSIF #Start < 0 THEN #PaFE := 4; //Startwert kleiner 0 ELSE
#PaFE := 0; END_IF; //Daten kopieren, wenn Fehler = 0 IF #PaFE = 0 THEN
(*****String Laenge in Ausgabestring schreiben***** //BZ_Hilfsvariable ist
der gesicherte Beginn vom unveraenderten Zeiger //Zeiger setzen auf Byte 0, dies entspricht der String
Laenge #p_DATA_OUT.BZ := (#dBZ_Hilfsvariable AND dw#16#FFFF0000) OR #dBZ_Hilfsvariable
(*+ 0*); //Header des OUTPUT schreiben #erg_SFC20 := BLKMOV(SRCBLK :=
#bHeader_Str_Laenge, DSTBLK => #DATA_OUT_T);
(*****
*****Belegte Laenge in Ausgabestring schreiben***** //BZ_Hilfsvariable
ist der gesicherte Beginn vom unveraenderten Zeiger //Zeiger setzen auf Byte 1, dies entspricht der
belegten String Laenge #p_DATA_OUT.BZ := (#dBZ_Hilfsvariable AND dw#16#FFFF0000) OR
INT_TO_DWORD(DWORD_TO_INT(#dBZ_Hilfsvariable) + 1 * 8); //Header des OUTPUT
schreiben #erg_SFC20 := BLKMOV(SRCBLK := #bHeader_Str_Inhalt, DSTBLK => #DATA_OUT_T);
(*****
***** //Zeiger anpassen
INPUT #p_DATA_IN.DataCount := BYTE_TO_WORD(#bHeader_Str_Inhalt);
#p_DATA_IN.BZ := (#p_DATA_IN.BZ AND dw#16#FFFF0000) OR
INT_TO_DWORD(DWORD_TO_INT(#p_DATA_IN.BZ) + #Start * 8); //Zeiger anpassen OUTPUT
#DATA_OUT_T := #DATA_OUT; #p_DATA_OUT.DataCount := #p_DATA_IN.DataCount;
#p_DATA_OUT.BZ := (#dBZ_Hilfsvariable AND dw#16#FFFF0000) OR
INT_TO_DWORD(DWORD_TO_INT(#dBZ_Hilfsvariable) + 2 * 8); //Daten kopieren
#erg_SFC20 := BLKMOV(SRCBLK := #DATA_IN_T, DSTBLK => #DATA_OUT_T);
END_IF;END_REGION

```

