

FANUC 机器人

Socket 坐标数据拆分程序

修订	姓名	日期
编制	胡欣琪	2022/6/8
审核		
批准		

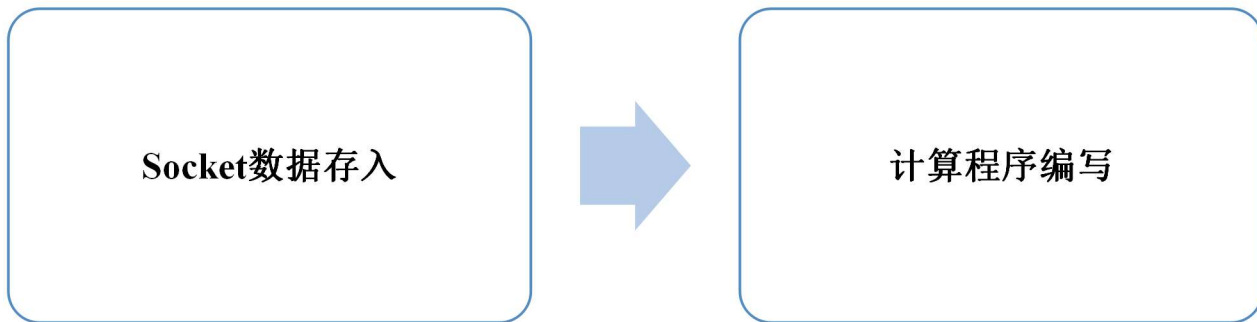
版本	更新内容	日期	姓名
V1.0	首次发布		

目录

1、概述.....	3
2、程序编写.....	3
2.1 数据存入.....	3
2.2 拆分程序编写.....	5
3、常见报警.....	6
3.1 INTP-213 用户报警.....	6
4、附录.....	7

1、概述

使用 Socket 标准通讯 pc 程序时，若上位机发送的数据为字母和数字混合形式，需要机器人端进行拆分。本手册提供一种字符串寄存器的拆分思路，拆分后的数据机器人端使用起来更为简单。



2、程序编写

2.1 数据存入

使用标准 pc 程序将上位机的数据存入 SR[1]寄存器。

示例：

```
CALL DATA_RECV(3,1,'C1')
```

<p>DATA_RECV</p>	<p>data_recv.pc</p>	<p>机器人接收相机发送的字符串数据</p> <p>输入参数:</p> <p>参数 1 (int: 1-4) int: 1-4 代表四种数据格式 1: 机器人接收位置坐标, 格式为 Model_ID, X, Y, R, 四个数据为一组, 最多 8 组 2: 机器人接收 Pass/Fail 信号, 格式为 1 或 0 3: 机器人接收 OCR 字符串, 最多 20 个字符 4: 机器人接收测量数据, 最多 20 个数据</p> <p>参数 2 (int) 根据参数 1, 参数意义范围不同 1: int 代表的位置寄存器序号, 若有多组则每次调用存放一组, 并在 int 代表的数字寄存器 Num_Found 内存放剩余数据个数, int+1 的数字寄存器存放 Model_ID, int 应在 100 以内 2: int 代表结果存放的数字寄存器序号, int 应在 100 以内 3: int 代表结果存放的字符串寄存器序号, int 应在 25 以内 4: int 代表结果存放的数字寄存器起始序号, int 应在 80 以内</p> <p>参数 3 (string: C1-C8 或 S1-S8) string: 机器人作为客户端或服务端的 Tag 号 (必须大写)</p> <p>举例: CALL DATA_RECV(1, 1, 'C1') 机器人作为客户端, 接收相机位置数据, 并将结果存放于 PR[1], Num_Found 存放于 R[1], Model_ID 存放于 R[2]。</p> <p>输出参数: 根据参数 1, 输出不同参数 1: 输出位置数据, Num_Found, Model_ID 2: 输出 Pass/Fail 信号 3: 输出 OCR 字符串 4: 输出测量结果数据</p>
------------------	---------------------	---

2.2 拆分程序编写

使用前在 SR[25]中提前写入数据间的分隔符，如英文逗号“，”。具体程序如图所示：

```

1: !SR[25]写入分隔符；
2: !SR[1]写入Socket数据；
3: SR[2]=SR[1] ；
4: R[1:Length]=STRLEN SR[1]；
5: R[50]=2 ；
6: R[51]=12 ；
7: R[52]=13 ；
8: R[12:STRT1]=1 ；
9: ；
10: LBL[100]；
11: IF R[50]>8,JMP LBL[999]；
12: R[R[50]]=FINDSTR SR[2],SR[25]；
13: IF R[R[50]]=0,JMP LBL[200]；
14: R[R[52]]=R[R[51]]+R[R[50]] ；
15: ；
16: ；
17: R[1:Length]=R[1:Length]-R[R[50]] ；
18: R[21]=R[R[50]]+1 ；
19: SR[2]=SUBSTR SR[2],R[21],R[1:Length]；
20: R[R[50]]=R[R[50]]-1 ；
21: R[50]=R[50]+1 ；
22: R[51]=R[51]+1 ；
23: R[52]=R[52]+1 ；
24: JMP LBL[100]；
25: ；
26: LBL[200]；
27: //PAUSE；
28: R[R[50]]=R[1:Length] ；
29: FOR R[99]=2 TO R[50]；
30: R[100]=R[99]+10 ；

31: R[101]=R[99]+20 ；
32: SR[R[99]]=SUBSTR SR[1],R[R[100]],R[R[99]]；
33: ENDFOR；
34: ；
35: END；
36: ；
37: LBL[999]；
38: UALM[1]；

```

运行结果如图所示：

```

SR[ 1:Offset Pos ]=C, 324. 45, 436. 56, 745. 36, 671. 01, 567. 00
SR[ 2: ]=C
SR[ 3: ]=324. 45
SR[ 4: ]=436. 56
SR[ 5: ]=745. 36
SR[ 6: ]=671. 01
SR[ 7: ]=567. 00
SR[ 8: ]=
SR[ 9: ]=

```

3、常见报警

3.1 INTP-213 用户报警

故障现象：Socket 字符串拆分失败

故障原因：Socket 数据数量大于 8 个

处理方案：

分段发送，多次拆分即可

4、附录
