事件驱动子程序在进口设备功能改进中的应用

赵红淑
南京高速齿轮制造有限公司　江苏南京　211100

摘要：介绍了西门子数控系统中的PROG_EVENT事件驱动子程序的应用。首先介绍了事件驱动编程的基本概念和原理，然后详细讲解了西门子数控系统中的PROG_EVENT事件驱动子程序的使用方法和应用案例，最后结合实际案例介绍了如何扩大其功效帮助调试设备以及指导解决在进口设备中遇到的问题。
关键词：事件驱动编程；西门子；数控系统；PROG_EVENT；子程序

1　序言

事件驱动编程是一种常见的编程方法，其基本原理是通过事件来触发程序的执行。在事件驱动编程中，程序会等待某个事件的发生，一旦事件发生，程序就会执行相应的操作。事件常见的是“程序启动”“程序停止”“通道复位”等，系统在这些事件发生时可以调用固定的子程序去做处理，这个子程序名为“PROG_EVENT.SPF”[1]。

事件驱动编程的优点是可以提高程序的响应速度和灵活性。由于程序不需要一直轮询某个状态，而是等待事件的发生，因此可以节省系统资源。此外，事件驱动编程还可以实现多任务并发执行，提高程序的效率。

2　PROG_EVENT事件驱动子程序的使用方法

2.1　如何调用事件驱动的子程序

“PROG_EVENT.SPF”就是被执行的子程序，如何才能正确调用呢？首先在MD20108 $MC_PROG_EVENT_MASK设定调用的事件，每个Bit都代表某个事件，比如当Bit0=1时，表示每次执行程序启动时后台都会调用“PROG_EVENT.SPF”[2]。事件调用参数说明如图1所示。

为了在PROG_EVENT.SPF程序中区分不同的事件，系统提供了变$P_PROG_EVENT：$P_PROG_EVENT=1 是“程序启动”事件；$P_PROG_EVENT=2 是“程序结束”事件；$P_PROG_EVENT=3 是“通道复位”事件；$P_PROG_EVENT=4 是“系统上电”事件；$P_PROG_EVENT=5 是“程序段搜索动作”事件。



图1　事件调用参数说明

MD11620 $MN_PROG_EVENT_NAME 是被事件触发的子程序的程序名，默认的程序名是PROG_EVENT.SPF，但其实不是必须用这个程序名PROG_EVENT.SPF。

另外，MD11622 $MN_PROG_EVENT_PATH 表示这个子程序放在那里。MD11622=0：表示子程序放在/_N_CMA_DIR目录下，即制造商循环目录；MD11622=1：表示子程序放在/_N_CUS_DIR目录下，即用户循环目录；MD11622=2：表示子程序放在/_N_CST_DIR目录下，即标准循环目录；MD11622=3：首先查找/_N_CUS_DIR，再查找/_N_CMA_DIR，最后查找/_N_CST_DIR。这里可以看出用户目录优先。子程序名和存放路径设置如图2所示。

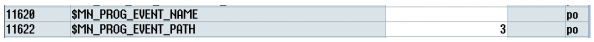


图2　子程序名和存放路径设置

2.2　子程序PROG_EVENT.SPF介绍

一般标准循环里可以看到PROG_EVENT.

SPF程序在开头分别调用了CYCPE1US.SPF（预留给用户的，用户事件触发子程序的程序名）和CYCPE1MA.SPF（预留给机床厂商的，用户事件触发子程序的程序名），如图3所示。

在PROG_EVENT.SPF程序快结束的地方，又分别调用了CYCPE_US.SPF（预留给用户的，用户事件触发子程序的程序名）和CYCPE_MA.SPF（预留给机床厂商的，用户事件触发子程序的程序名），如图4所示。



图3　程序内调用子程序1

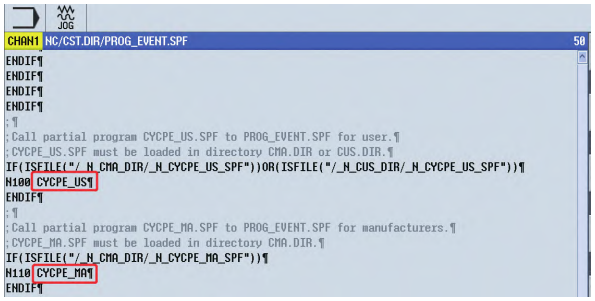


图4　程序内调用子程序2

至此，系统默认用名为PROG_EVENT.SPF的子程序来处理一些事件，作为一个标准循环程序，本身就包含了一些西门子定义好的内容，如果机床厂商也想针对某些事件发生时执行一些特定动作，那就要把程序段写在CYCPE1MA.SPF或CYCPE_MA.SPF中，这是因为PROG_EVENT.SPF执行时会调用这两个子程序。用户也可以按照这个思路去添加。

3　应用案例

3.1　案例一

改进设备：西班牙进口的大型龙门加工中心。要实现的目标：要求在将RMP600测头换刀到主轴后，各几何轴的最大移动速度自动降低至原速度的1/2。

在前面介绍的基础上，发现制造商做了如下的程序设定（程序调用关系见图5）。

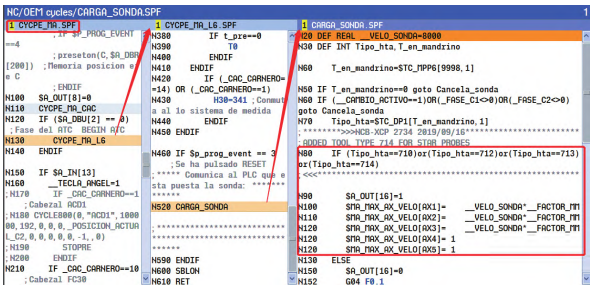


图5　程序调用关系

1）在CYCPE_MA.SPF中调用CYCPE_MA_L6，调用条件是开始ATC（自动换刀）。

2）在CYCPE_MA_L6里，$P_PROG_EVENT=3“通道复位”时再调用CARGA_SONDA。

3）在CARGA_SONDA里，设定判断条件程序如下。

Tipo_hta=$TC_DP1[T_en_mandrino,1]

IF (Tipo_hta==710)or(Tipo_hta==712)or(Tipo_hta==713)or(Tipo_hta==714)

即通过判断主轴刀具的$TC_DP1[ ]刀具类型，测头的刀具类型只能是710、712、713和714，测头所属刀具类型参数如图6所示，从而实现对速度参数的重新赋值。这里用到了局部变量__VELO_SONDA和全局变量__FACTOR_MM，通过修改后两者的赋值就可以达到修改最大速度的目的。

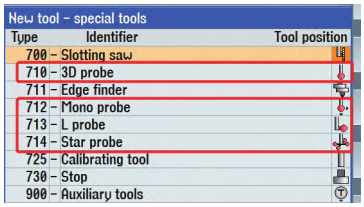


图6　测头所属刀具类型参数

3.2　案例二

改进设备：意大利进口的大型卧式加工中心。要实现的目标：增加一个垂直补偿表格用于高精度加工时YZ轴垂直补偿，总是无法一直生效，排查原因。

按照标准设定，新建了垂直补偿的4号表格，并将41300[4]激活（垂直补偿表格设置见图7），NC读入补偿表格后，再通过激活OUTPUT轴的MD32710使得补偿生效[3]。

可是这个值总感觉没能一直生效，后来发现每次按过RESET后补偿值就失效了，在“轴诊断”（见图8）里可以看出。

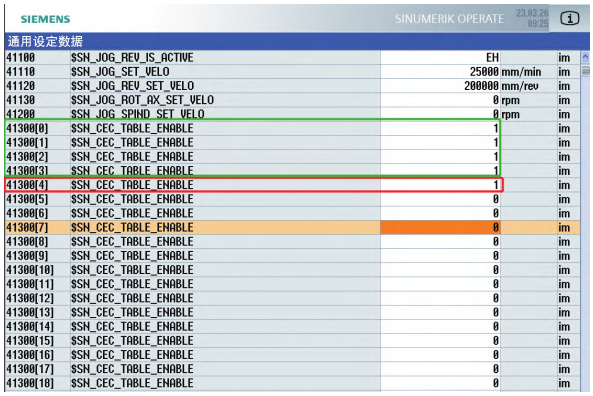


图7　垂直补偿表格



图8 数控系统“轴诊断”

所以按“RESET”是个关键信息，自然而然想到事件驱动子程序PROG_EVENT.SPF，顺藤摸瓜我们发现CYCPE_MA.SPF中在复位RESET时会调用子程序“PK_VERAC”，具体该子程序调用如图9所示。

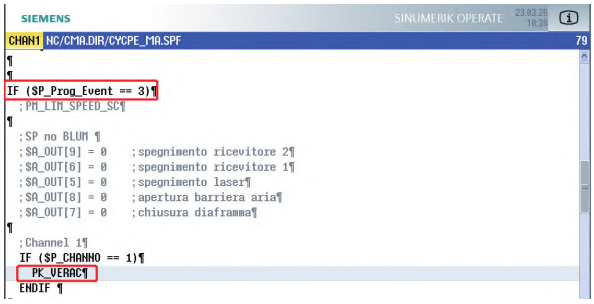


图9　子程序调用

打开后可以发现制造商在其中做了一段关于垂直补偿表格生效的初始化设定，显然$SN_CEC_TABLE_ENABLE[4]已经被强制置0了（垂直表格初始化设定见图10），因为每次按完RESET后都会调用该子程序，达到初始化的目的，所以要想启用该表格必须人为地设置为1，当然修改前还要阅读相关程序、考虑制造商设置背后的一些应用场合，不要挤占制造商的设计思路。此处只要修改这里的设

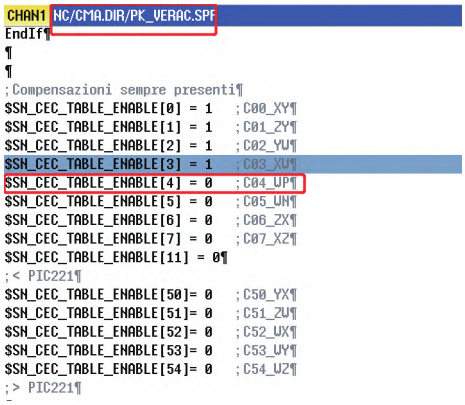


图10　垂直表格初始化设定

定，再进行一开始的补偿设定即可。

通过以上两个应用案例的介绍，可以对事件驱动子程序有更深刻的理解。最近公司有个新的改造项目，需要给设备增做大型刀具自动更换的功能，刀具类型属于镗刀，镗刀直径1m左右，在主轴定位时总是会报警，很显然，主轴的伺服参数不匹配这么大的刀具进行定位，借助于事件驱动子程序，是可以予以解决的。

4　结束语

PROG_EVENT子程序与其他程序相比，具有如下特点。

（1）高效性　在事件驱动的情况下，程序只在需要时才会运行，从而节省了系统资源，提高了程序运行效率。

（2）灵活性　PROG_EVENT子程序可以在任何时候按照事件去调用，而不需要按照预定流程执行。

（3）可维护性　由于PROG_EVENT程序是独立的，所以对其他程序的修改不会对其造成影响，也更容易进行程序的维护。

事件驱动子程序的应用场合比较广泛，本质上就是一个异步中断子程序，特别适合后期的改造、制造企业想法的实现等。

参考文献：

[1] 西门子有限公司. SINUMERIK 840D sl/828基本功能[S]. 2015.

[2] 西门子有限公司. SINUMERIK 840D sl/828扩展功能[S]. 2015.

[3] 张泰华，张海斌. SINUMERIK 840D sl数控系统调试与应用[M]. 北京：机械工业出版社，2015.

MW 20240321