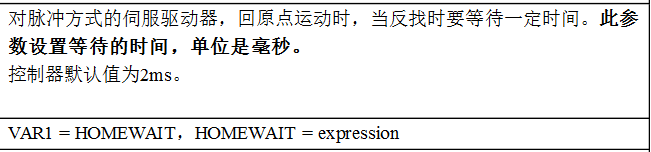
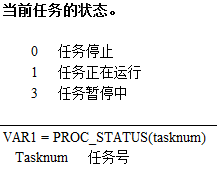
在看“4轴带视觉 - Zhmi版”这个例程 第一个界面的启动，停止，暂停，归零这四个功能。以及HMI界面的制作。和HMI寄存器的使用



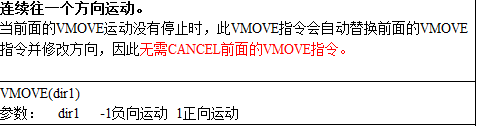
**手动操作**

手动和手动停止

PROC\_STATUS



Vmove

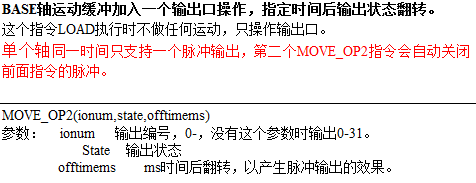


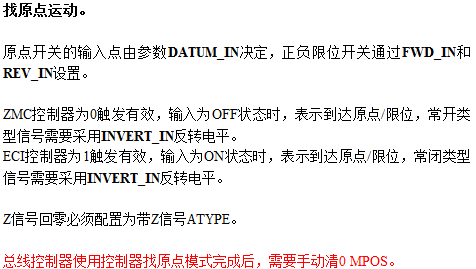
RAPIDSTOP

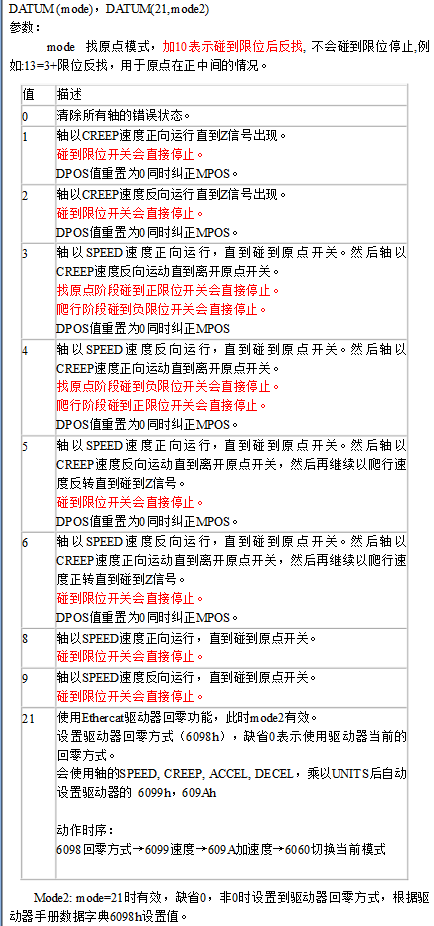


**错误信息**

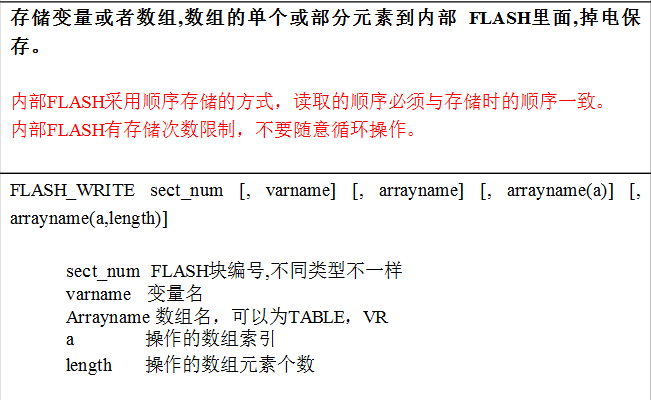
move\_op2

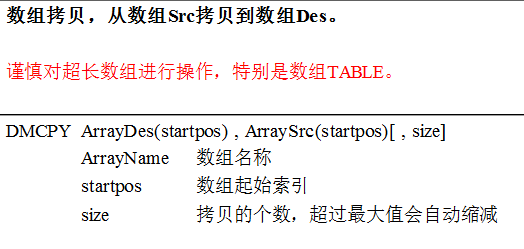






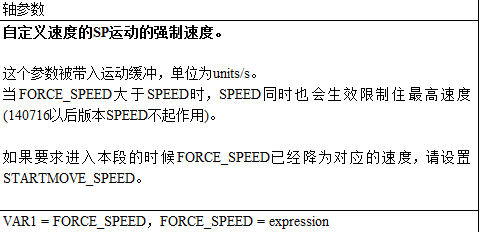
轴参数设置 保存



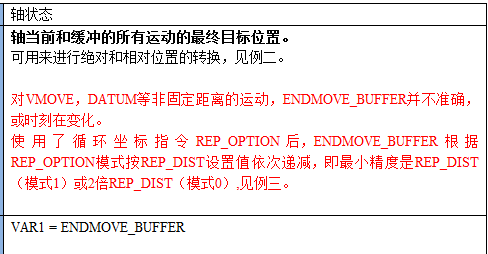




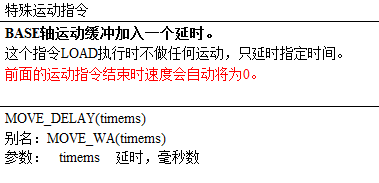
直线



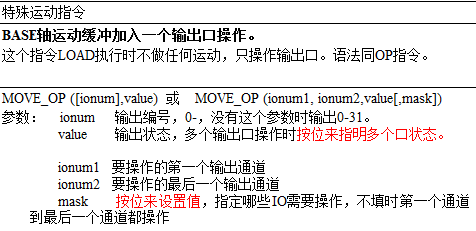
三点圆弧



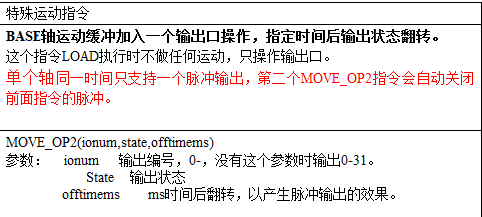
延时



输出



输出复位



0起始速度 lspeed

1运行速度 speed

2加速度 accel

3减速度 decel

4S曲线 sramp

5复位速度 speed

6反找速度 creep

7示教速度

8回零方式 datum

9反找等待 HOMEWAIT

10每圈脉冲数

11螺距

12负向软限位 RS\_LIMIT

13正向软限位 FS\_LIMIT

14原点IN datum\_in

15反转 INVERT\_IN

16正限位IN fwd\_in

17反转 INVERT\_IN

18负限位IN rev\_in

19反转 INVERT\_IN

20报警IN alm\_in

21反转 INVERT\_IN

22使能OP op

23报警清除op

24预留

与库的结合

Basic

Speed Int32 ZAux\_Direct\_SetSpeed(IntPtr handle, int iaxis, float fValue);

vr寄存器 Int32 ZAux\_Direct\_SetVrf(IntPtr handle, int vrstartnum, int numes, float[] pfValue)

Int32 ZAux\_Direct\_GetVrf(IntPtr handle, int vrstartnum, int numes, float[]pfValue);

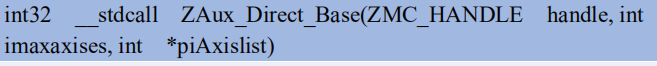
move\_mark Int32 ZAux\_Direct\_SetMovemark(IntPtr handle, int iaxis, int iValue);

wait idle

MOVE\_RESUME

move\_pause

Base



RAPIDSTOP Int32 ZAux\_Direct\_Rapidstop(IntPtr handle, int imode);

HOMEWAIT Int32 ZAux\_Direct\_SetHomeWait(IntPtr handle, int iaxis, int fValue);

Datum Int32 ZAux\_Direct\_Single\_Datum(IntPtr handle, int iaxis, int imode);

Idle Int32 ZAux\_Direct\_GetIfIdle(IntPtr handle, int iaxis, ref int piValue);

Vmove Int32 ZAux\_Direct\_Single\_Vmove(IntPtr handle, int iaxis, int idir);

Cancel Int32 ZAux\_Direct\_Single\_Cancel(IntPtr handle, int iaxis, int imode);

Move Int32 ZAux\_Direct\_Single\_Move(IntPtr handle, int iaxis, float fdistance);

Moveabs Int32 ZAux\_Direct\_Single\_MoveAbs(IntPtr handle, int iaxis, float fdistance);

Dpos Int32 ZAux\_Direct\_SetDpos(IntPtr handle, int iaxis, float fValue);

Mpos Int32 ZAux\_Direct\_SetMpos(IntPtr handle, int iaxis, float fValue);

RAPIDSTOP Int32 ZAux\_Direct\_Rapidstop(IntPtr handle, int imode);

Base

Atype Int32 ZAux\_Direct\_SetAtype(IntPtr handle, int iaxis, int iValue);

Units Int32 ZAux\_Direct\_SetUnits(IntPtr handle, int iaxis, float fValue);

Lspeed Int32 ZAux\_Direct\_SetLspeed(IntPtr handle, int iaxis, float fValue);

Accel Int32 ZAux\_Direct\_SetAccel(IntPtr handle, int iaxis, float fValue);

Decel Int32 ZAux\_Direct\_SetDecel(IntPtr handle, int iaxis, float fValue);

Sramp Int32 ZAux\_Direct\_SetSramp(IntPtr handle, int iaxis, float fValue);

Creep Int32 ZAux\_Direct\_SetCreep(IntPtr handle, int iaxis, float fValue);

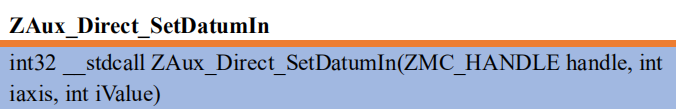
RS\_LIMIT



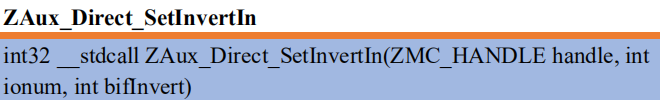
FS\_LIMIT



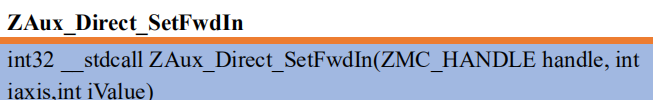
datum\_in



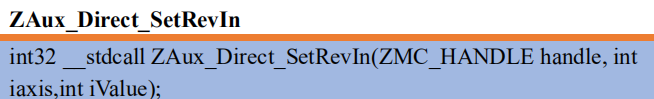
INVERT\_IN



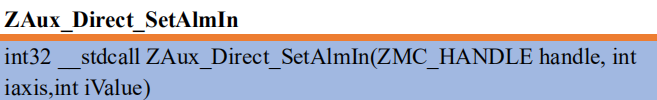
fwd\_in



rev\_in



alm\_in



Op Int32 ZAux\_Direct\_MoveOp(IntPtr handle, int iaxis, int ioutnum, int ivalue);

FLASH\_WRITE

u\_state

File

DMCPY

ZINDEX\_ARRAY

flash\_read

move\_op2

DMDEL

wa

LOADED Int32 ZAux\_Direct\_GetLoaded(IntPtr handle, int iaxis, ref int piValue);

Axisstatus Int32 ZAux\_Direct\_GetAxisStopReason(IntPtr handle, int iaxis, ref int piValue);

read\_bit2

STRFIND

DMDEL

force\_speed



Movesp Int32 ZAux\_Direct\_MoveSp(IntPtr handle, int imaxaxises, int[] piAxislist, float[] pfDposlist);

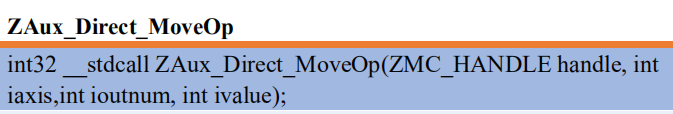
Moveabssp Int32 ZAux\_Direct\_MoveAbsSp(IntPtr handle, int imaxaxises, int[] piAxislist, float[] pfDposlist);

MSPHERICALSP Int32 ZAux\_Direct\_MSphericalSp(IntPtr handle, int imaxaxises, int[] piAxislist, float fend1, float fend2, float fend3, float fcenter1, float fcenter2, float fcenter3, int imode, float fcenter4, float fcenter5);

ENDMOVE\_BUFFER

move\_delay



move\_op 

Hmi

ZINDEX\_CALL