

RobotAnno 控制器指令集文档

1. 串口指令说明

1.1 串口模式

串口参数：波特率 115200；数据位 8；停止位 1。

指令格式：

控制指令为 16 进制的 0X10(空闲模式),0X11(文件模式),0X12(回零模式),0X13(运行模式),0X14(调试模式),0X15(复位模式);0X05 为询问模式；

运行指令全部以'GXX'开头，以回车换行结束；

示教指令以回车换行结束；

在 180521 及以上系统版本,当参数文件里设置_flagSoftRst=1，任意情况串口 1 都可以发 0X18 使输出口 o14 输出,可根据此功能 o14 接继电器控制 NRST 重启(急停)

0X05: ==询问模式==

控制卡在任意工作模式下(文件模式除外)，串口发送 0X05 可以询问控制卡的当前工作模式。返回值有 0X10, 0X12, 0X13, 0X14, 0X15。

0X10: ==进入空闲模式==

控制卡在空闲模式时(0X10):为空闲模式不变。

在文件模式时，在文件操作指令运行完后发送 0X10 生效。

在回零模式时，机器人停止运动(发送 0X30)后发送 0X10 生效。

在运行模式时，机器人停止运动(发送 0X30)后发送 0X10 生效。

在复位模式时，复位完成制动退出到空闲模式。开机时为空闲模式。

0X11: ==进入文件模式==

下载运行文件在发送 0X11 后选中文件点击“发送文件”即可。其中，文件中第一行为

"wr=ST"是运行文件；

"WR=INI"是参数文件；

"WR=oq"是回零文件；

"WR=T_0"是默认运行文件，等同于"wr=ST"；

"WR=T_1"是运行 1 文件；

"WR=T_2"是运行 2 文件；

...

"WR=T_120"是运行 120 文件；

第二行为文件名称；第三行为文件长度，长度为除去前面三行的文件大小。

读取运行文件在发送 0X11 后，发送"WR=R,###"("###"为读取内容长度)即可接收到运行文件。

读取参数文件在发送 0X11 后，发送"WR=P,###"("###"为读取内容长度)即可接收到参数文件。

读取参数文件在发送 0X11 后，发送"WR=Q,###"("###"为读取内容长度)即可接收到回零文件。

读取参数文件在发送 0X11 后，发送"WR=H_0,###"("###"为读取内容长度)即可接收到运行 0 文件。

读取参数文件在发送 0X11 后, 发送"WR=H_1,###"("###"为读取内容长度)即可接收到运行 1 文件。

...

读取参数文件在发送 0X11 后, 发送"WR=H_120,###"("###"为读取内容长度)即可接收到运行 120 文件。

0X12: ==电机回零模式==

进入回零模式后, 系统自动运行文件内容与串口实时发送的指令。暂停为 0X30, 继续为 0X12, 退出为 0X10

指令可从运行文件中获得, 也可接受串口获得。

运行完文件内容后, 自动退出空闲模式

0X13: ==进入运行模式==

进入运行模式后, 系统自动运行文件内容与串口实时发送的指令。暂停为 0X30, 继续为 0X13, 退出为 0X10。

指令可从运行文件中获得, 也可接受串口获得。

运行完文件内容后, 自动从头开始运行。

0X14: ==进入调试模式==

发送 0X05 为询问模式。

发送 0X10 为退出模式。

发送"0"为停止, 并发送机器各轴角度。

发送"SVON=1"为伺服使能。

发送"SVON=0"为伺服不使能。

发送"SVBM=1"为伺服不抱闸。

发送"SVBM=0"为伺服抱闸。

发送"CLR=ST_PUL"为控制卡的位置清零。

发送"CLR=SV_ALM"为清除驱动器报警。

发送"J#+ "为第#轴角度加运动(#=1, 2, 3, 4, 5, 6)。

发送"J#-"为第#轴角度减运动。

发送"J#0"为第#轴角度停止运动。

发送"J#<"为第#轴角度(@ 0° ~ -90°)重复运动。

发送"J#>"为第#轴角度(@ 0° ~ 90°)重复运动。

发送"P VE=####"为改变运行速度为####个脉冲/秒。

发送"P AC=####"为改变加速度为####个脉冲/平方秒。

发送"P DE=####"为改变减速度为####个脉冲/平方秒。

发送"G00 J#=####,..."为使机器人关节走到相应角度。

发送"G07 **=####"为改变机器人指定参数。

TX+:直角坐标 X+;

TX-:直角坐标 X-;

TY+:直角坐标 Y+;

TY-:直角坐标 Y-;

TZ+:直角坐标 Z+;

TZ-:直角坐标 Z-;

TA+:直角坐标 A+;
 TA-:直角坐标 A-;
 TB+:直角坐标 B+;
 TB-:直角坐标 B-;
 TC+:直角坐标 C+;
 TC-:直角坐标 C-;
 P CYCP1=800000:J1 轴细分为 800000 脉冲每周

 P J1=24000 给 J1 赋值

 P=VE 查询 VE 值 可查的有 VE,AC,DE,CYCP1,...,J1, ...,H0, D1, H2...

0X15: ==进入复位模式==
 使机器回到零位。

2.按键控制说明

开机时，按"开始"按键为机器人到运行模式运行。
 运行模式下，在运行时，按"停止"按键机器暂停。
 运行模式下，在停止时，按"开始"按键，机器继续运行。
 运行模式下，在停止时，按"停止"按键不松手再按"开始"按键，机器复位并退出空闲模式。

指令说明：

//=====

G0n=GGN 如 G06 O=P1.1 等同 GG6 O=P1.1
 点对点指令 G00, G20
 G00 J1=0 J2=0 J3=-90 J4=0 J5=-90 J6=0 //带加减速 绝对关节位置指令
 G00 J1=VXX J2=0 J3=VXX J4=VXX J5=-90 J6=0 //带加减速 绝对关节位置指令
 G00 ... J2'4 J3'VXX ... //带加减速 相对关节位置指令
 G00 ... J2'4 J3'VXX J4=VXX J5=-90... //J2 为
 G20 X=300 Y=100 Z=500 A=0 B=180 C=0 D=0 //带加减速
 G20 X=300 Y=VXX Z=VXX A=VXX B=180 C=VXX D=0 //带加减速
 G30 EQ01 J1=UXXXX,J2=UXXXX,J3=UXXXX 读取存储器内容 G30 EQ01 J1=U30100 J2=U30120
 J3=U30140 J4=U30160
 G30 EQ02 U3060=XX.XX 写存储器内容 G30 EQ02 U30500=10
 G30 EQ03 ASK=UXXXX 查询存储器内容
 G30 EQ04 Vxx=UXXXXXX
 G30 EQ05 UXXXXXX=Vxx
 G40 X=300 Y=100 Z=500 A=0 B=180 C=0 D=0 //无加减速

直线指令 G01 G21

G01 J1=0 J2=0 J3=-90 J4=0 J5=-90 J6=0
 G21 X=300 Y=100 Z=500 A=0 B=180 C=0 D=0

G21 X=VXX Y=VXX Z=500 A=0 B=VXX C=0 D=0
G41 X=300 Y=100 Z=500 A=0 B=180 C=0 D=0 //带加减速
G41 X=300 Y=VXX Z=VXX A=0 B=180 C=0 D=0 //带加减速

圆弧指令 G02 G03 G04 G22 G23 G06 DEGREE=ARC(或者度数)
当 DEGREE=ARC 走的是三点弧线

当前点: G21 X=200 Y=0 Z=200 A=-180 B=150 C=0 D=0
第二点: G22 X=300 Y=100 Z=200 A=-180 B=150 C=90 D=0
第三点: G23 X=400 Y=0 Z=200 A=-180 B=150 C=300 D=0
圆弧运行: G06 DEGREE=300

//=====

G06 指令

G06 T=500 延时 500 毫秒

G06 t=VXX 延时 VXX 内容毫秒(例如:G08 MOV V400=#2000 G06 t=V400 为延时 2000 毫秒)

G06 I=P1.1 等待 P1 为高电平

G06 I=P2.0 等待 P2 为低电平

G06 O=P1.1 使输出口 P1 为高电平 (与输入口不同端口)

G06 O=P2.0 使输出口 P2 为低电平

G06 SCAN=I 读取输入端口值, 存于 V144-V150 单元中

G06 SCAN=O 读取输出端口值, 存于 V160-V166 单元中

G06 SCAN=RTC 读取系统时钟值, 存于 V176-V179 单元中

G06 DEGREE=ARC 机器人走三点圆弧

G06 DEGREE=35.2 机器人走 35.2 度

G06 REPOS=J# J#轴从角度增加方向找 HOME#感应器(回到低电平)

G06 REPOS=-J# J#轴从角度减小方向找 HOME#感应器(回到低电平)

G06 REPOS=J1 J1 轴从角度增加方向找 HOME0 感应器(回到低电平)

G06 REPOS=-J1 J1 轴从角度减小方向找 HOME0 感应器(回到低电平)

G06 REPOS=JH# J#轴从角度增加方向找 HOME#感应器(回到高电平)

G06 REPOS=-JH# J#轴从角度减小方向找 HOME#感应器(回到高电平)

G06 REPOS=JH1 J1 轴从角度增加方向找 HOME0 感应器(回到高电平)

G06 REPOS=-JH1 J1 轴从角度减小方向找 HOME0 感应器(回到高电平)

//=====

G07 指令

G07 VE=250000 速度是 250000 脉冲每秒

G07 AC=250000 加速度是 250000 脉冲每秒平方

G07 DE=250000 减速度是 250000 脉冲每秒平方

G07 VPP=250000 最高速度 250000 脉冲每秒

G07 VP=20 速度 VE 是最高速度的 20%, $VE=VPP*VP*0.01$

G07 vp=VXX 速度 VE 是最高速度的 VXX 里的内容, $VE=VPP*VXX$ 单元里的值*0.01

G07 _h0=xx 高度改变

G07 RCM=1 打印运行指令, 0 不打印运行指令

G07 GCM=1 示教时以直角坐标输出, 0 为关节坐标输出

```

G07 MARKPOS_HERE X1=0 Y1=0 X2=100 Y2=100 模板 MARK 点
G07 MARKPOS X1=10 Y1=10 X2=110 Y2=110 匹配 MARK 点
G07 UNIT=1.0 插补精度（单位 mm）
G07 P J#=XXXX 标定 J#轴角度
//=====
G08 指令
G08 XXXX:标签 标签后面禁止加注释，标签长度需小于 15 个字节
G08 ACALL XXXX 调用 XXXX 标签到 G08 END
G08 END 调用结束
G08 AJMP XXXX 跳转到 XXXX 运行
G08 FLTAB=# 文件跳转到 T_#号文件运行；

G08 IF VXX>=VXX ACALL XXXX 整数比较 立即数为# 比较>,<=,==,!=,>,<;(G08 IF VXX>=VXX
AJMP XXXX)
G08 IF_ELSE VXX>=VXX?ACALL XXXX:ACALL XXXX 真值运行冒号前，假值运行冒号后 ACALL 为
调用，AJMP 为跳转（G08 IF_ELSE VXX>=VXX?AMMP XXXX:AJMP XXXX）
G08 IFF 浮点比较
G08 IF_ELSEF 浮点比较

G08 MOV VXX=#XX 整数传送(立即数 G08 MOV V10=#12 )(单元 G08 MOV V10=V12)
G08 MOVF VXX=#XX 整数传送(立即数 G08 MOVF V10=#12.5)(单元 G08 MOVF V10=V12)
G08 PRINT VXX 打印 VXX 的值，整数形式
G08 PRINTF VXX 打印 VXX 的值，浮点形式

G08 INT VXX 把浮点转换为整数*****
G08 FLOAT VXX 把整数转换为浮点*****

G08 ADD VXX=VXX+VXX 整数 立即数为#
G08 SUBB VXX=VXX-VXX 整数 立即数为#
G08 MUL VXX=VXX*VXX 整数 立即数为#
G08 DIV VXX=VXX/VXX 整数 立即数为#
G08 SQRT VXX=VXX 整数 立即数为#

G08 ADDF VXX=VXX+VXX 浮点 立即数为#
G08 SUBBF VXX=VXX-VXX 浮点 立即数为#
G08 MULF VXX=VXX*VXX 浮点 立即数为#
G08 DIVF VXX=VXX/VXX 浮点 立即数为#
G08 SQRTF VXX=VXX 浮点 立即数为#

G08 STO 程序自动暂停*****170817*****
G08 EXIT 退出运行进入空闲模式*****170817*****
//=====
G09 指令
G09 COPYRIGHT 询问系统 YN

```

G09 COM2=XXXX COM2 口发送 XXXX 字符

例如：G09 COM2=G00 J1=10 J2=10 J3=-90 J4=0 J5=0 J6=0

此控制器发出指令到另外一个控制器 COM1,

G09 COM2 10H 串口 2 发出 0X10
G09 COM2 12H 串口 2 发出 0X12
G09 COM2 13H 串口 2 发出 0X13
G09 COM2 14H 串口 2 发出 0X14
G09 COM2 15H 串口 2 发出 0X15
G09 COM2 18H 串口 2 发出 0X18

G09 ENC....与绝对值伺服电机连接是专用指令，参考回零文件示例;

//=====

VXX: 单元

#XX: 立即数

用户运算单元 V0-V127,V400-V511

系统已用运算单元 读取输入端口值，存于 V144-V150 单元中

读取输出端口值，存于 V160-V166 单元中

读取系统时钟值，存于 V176-V179 单元中

在运行模式下或者运行 G 代码时发送'0'为停止，继续时为本条指令；'!'也是停止，继续时为下一条指令；

STA 实时按键值实时存与 V180 单元中，按键值记忆存与 V183 单元中，V183 由软件清除

STO 实时按键值实时存与 V181 单元中，按键值记忆存与 V184 单元中，V184 由软件清除

RST 实时按键值实时存与 V182 单元中，按键值记忆存与 V185 单元中，V185 由软件清除

V186: 输入口 i03 或 i04 作指令取消开启，V186=0 时不启用(默认)，V186=1 时启用

V187: 限位与伺服报警总开关 V187=1 时不启用(默认)，V187=0 时启用

V188: 软限位停止总开关 V188=1 时不启用(默认)，V188=0 时启用

V189: 输入信号捕获总开关 V189=1 时不启用(默认)，V189=0 时启用

V143: 运行脉宽(值范围 0~400,自动修改)

V142: 脉宽阈值(值范围 4~400,人为修改)

V141: G41,G06 DGREE 平滑值(停止速度,人为修改)

V140: 速度定时值（自动更改）

端口对应寄存器值

(扫描时 G06 SCAN=I, V144~V159,V192~V207:灯亮值为 0, 灯灭值为 1)

i00:p0 v144

i01:p1 v145

i02:p2 v146

i03:p3 v147

i04:p4	v148
i05:p5	v149
i06:p6	v150
i07:p7	v151
i08:P8	v152
i09:P9	v153
i10:P10	v154
i11:P11	v155
i12:P12	v156
i13:P13	v157
i14:P14	v158
i15:P15	v159

HM0:P16	v192
HM1:P17	v193
HM2:P18	v194
HM3:P19	v195
HM4:P20	v196
HM5:P21	v197

LP0:P22	v198
LP1:P23	v199
LP2:P24	v200
LP3:P25	v201
LP4:P26	v202
LP5:P27	v203

ALM0:P28	v204
ALM1:P29	v205
ALM2:P30	v206
ALM3:P31	v207

O00:P0	v160
O01:P1	v161
O02:P2	v162
O03:P3	v163
O04:P4	v164
O05:P5	v165
O06:P6	v166
O07:P7	v167
O08:P8	v168
O09:P9	v169
O10:P10	v170
O11:P11	v171

O12:P12	v172
O13:P13	v173
O14:P14	v174
O15:P15	v175

(V256~V271:值为 1 捕获使能打开, V189=0 生效)

ENi00:p0	v256
ENi01:p1	v257
ENi02:p2	v258
ENi03:p3	v259
ENi04:p4	v260
ENi05:p5	v261
ENi06:p6	v262
ENi07:p7	v263
ENi08:P8	v264
ENi09:P9	v265
ENi10:P10	v266
ENi11:P11	v267
ENi12:P12	v268
ENi13:P13	v269
ENi14:P14	v270
ENi15:P15	v271

(V288~V303:捕获时长值, 单位毫秒, 需软件清零)

FLAGi00:p0	v288
FLAGi01:p1	v289
FLAGi02:p2	v290
FLAGi03:p3	v291
FLAGi04:p4	v292
FLAGi05:p5	v293
FLAGi06:p6	v294
FLAGi07:p7	v295
FLAGi08:P8	v296
FLAGi09:P9	v297
FLAGi10:P10	v298
FLAGi11:P11	v299
FLAGi12:P12	v300
FLAGi13:P13	v301
FLAGi14:P14	v302
FLAGi15:P15	v303

//=====

视觉示例

G08 CCD2:

G08 MOV V20=#0

G08 CCD2A:

G06 T=401//RCCD

G08 IF V20==#0 AJMP CCD2A

G08 END

在上位机收到 T=401 时，

上位机发送指令如：

G20 X=300 Y=200 Z=270 A=0 B=150 C=0 D=0 MOV V20=#1

G41 X=300 Y=200 Z=270 A=0 B=150 C=0 D=0 MOV V20=#1

//.....

输入信号示例

1,初始化，把使能打开 ENi07:p7 v263

G08 CSH:

G08 MOV V189=#0

G08 MOV V263=#1

G08 END

2,在运行中读出捕获 FLAGi07:p7 v295 值

G08 INSCAN:

G06 T=500

G08 IF V295>#400 ACALL ZHOUO14

G08 MOV V295=#0

G08 AJMP INSCAN

G08 END

3,捕获 v295 值大于 400 毫秒则调用 ZHOUO14

G08 ZHOUO14:

G06 O=P14.1

G06 T=400

G06 O=P14.0

G06 T=400

G06 O=P14.1

G06 T=400

G06 O=P14.0

G08 END

//=====

c 语言中 for,while 在 PFOP 上的实现

[1]例如:

int V0;

int V1=100;

for(V0=2,V0<V1,V0=V0+3)

{....}

.....

[1]在 PFOP 上的实现:

```

G08 MOV V0=#2
G08 MOV V1=#100
G08 AJMP FOR1
G08 LABEL1:
G08 ADD V0=V0+#3
....
G08 FOR1:
G08 IF_ELSE V0<V1?AJMP LABEL1: AJMP LABEL2
G08 LABEL2:
.....

```

[2]例如:

```

int V0;
int V1=100;
for(V0=2,V0<V1,V0=V0+3)
{
    ....
    if(V0>50){break;}
    ....
}
.....

```

[2]在 PFOP 上的实现:

```

G08 MOV V0=#2
G08 MOV V1=#100
G08 AJMP FOR1
G08 LABEL1:
G08 ADD V0=V0+#3
....
G08 IF V0>50 AJMP LABEL2
....
G08 FOR1:
G08 IF_ELSE V0<V1?AJMP LABEL1: AJMP LABEL2
G08 LABEL2:
.....

```

[3]例如:

```

int V0=2;
int V1=100;
while(V0<V1)
{
    V0=V0+1
    ....
}
.....

```

[3]在 PFOP 上的实现:

```
G08 MOV V0=#2
G08 MOV V1=#100
G08 AJMP FOR1
G08 LABEL1:
G08 ADD V0=V0+#1
....
G08 FOR1:
G08 IF_ELSE V0<V1?AJMP LABEL1: AJMP LABEL2
G08 LABEL2:
.....
```

[4]例如:

```
int V0=2;
int V1=100;
while(V0<V1)
{
    V0=V0+1
    ....
    if(V0>50){break;}
    ....
}
.....
```

[4]在 PFOP 上的实现:

```
G08 MOV V0=#2
G08 MOV V1=#100
G08 AJMP FOR1
G08 LABEL1:
G08 ADD V0=V0+#1
....
G08 IF V0>50 AJMP LABEL2
....
G08 FOR1:
G08 IF_ELSE V0<V1?AJMP LABEL1: AJMP LABEL2
G08 LABEL2:
.....
//=====
参数文件
FILE=INI      //文件路径
Parameter.ini  //文件名称
510           //文件长度（从第四行开始算到最后）
////DH_PARAMETER:  //DH 参数（单位：毫米；h 位长度，d 为距离）
```

```

_h0=203;    //可为虚拟值
_d1=80;
_h2=260;
_d3=80;
_h4=288;
_h5=64.5;
_d6=40;      //可为虚拟值
////MOTOR_DIR:    //马达方向
_dir1=-1;
_dir2=1;
_dir3=1;
_dir4=-1;
_dir5=1;
_dir6=-1;
////JOINT_PUL:    //机构细分
_j1pul=320000;    //机构一圈需 320000 个脉冲
_j2pul=326400;
_j3pul=326400;
_j4pul=320000;
_j5pul=320000;
_j6pul=320000;
////VELOCITY:
_ac=100000.0;    //加速度
_de=100000.0;    //减速度
_vpp=100000.0;    //满高速
_vp=10;    //速度程度
////MODE:
_pve=8.887    //未用
_getCodeMode=0;    //0 为示教为关节输出，1 示教为直角输出
_runCodeMode=1;    //1 为打印代码，0 为不打印代码
_prmTPUL=120;    //脉冲最大宽度调节阈值(4~400)(V142)
_prmVLIM=2000;    //G41 平滑值(V141)
//REPOS    //=====上电默认角度=====
_rePosJ1=0;
_rePosJ2=0;
_rePosJ3=-90;
_rePosJ4=0;
_rePosJ5=-90;
_rePosJ6=0;
//IRQ    //=====中断开关=====
_iIRQ=1    //感应器限位使能开关:    _iIRQ=1 使能关,_iIRQ=0 使能开,默认 1;
_sIRQ=1    //软限位使能开关:    _sIRQ=1 使能关,_sIRQ=0 使能开,默认 1;
_flagSoftRst=1    //串口重启控制器使能:    _flagSoftRst=1 使能开,_flagSoftRst=0 使能关,
默认 0;

```

```
_brkMotorEn=0          //抱闸使能: _brkMotorEn=1 使能开,_brkMotorEn=0 使能关,默认 1;
//LIMIT               //=====软件限位=====
_sLp0=175              //第 1 轴软件正限位角度
_sLn0=-175             //第 1 轴软件负限位角度
_sLp1=90               //第 2 轴....
_sLn1=-90              //第 2 轴....
_sLp2=30               //第 3 轴....
_sLn2=-180             //第 3 轴....
_sLp3=175              //第 4 轴....
_sLn3=-175             //第 4 轴....
_sLp4=120              //第 5 轴....
_sLn4=-120             //第 5 轴....
_sLp5=359              //第 6 轴....
_sLn5=-359             //第 6 轴....
```