

# 1FL6034-2AF21-1AB1现货西门子代理

|      |   |
|------|---|
| 产品名称 | 1FL6034-2AF21-1AB1现货西门子代理                               |
| 公司名称 | 湖南西控自动化设备有限公司   |
| 价格   | .00/件   |
| 规格参数 | 西门子:V90<br>电机:1FL6034-2AF21-1AB1<br>德国:2500线增量编码器带键槽带抱闸 |
| 公司地址 | 中国（湖南）自由贸易试验区长沙片区开元东路1306号开阳智能制造产业园（一期）4#栋301           |
| 联系电话 | 17838383235 17838383235                                 |

## 产品详情

1FL6034-2AF21-1AB1现货西门子代理

1FL6034-2AF21-1AB1现货西门子代理

1FL6034-2AF21-1AB1现货西门子代理

## 【干货】西门子S7-1200博途用LAD和SCL语言控制步进电机程序分析

西门子S7-1200PLC控制步进电机的梯形图程序

西门子S7-1200PLC控制步进电机的梯形图程序，也可以用结构化编程的思路来编程。

首先，新建一个FB，编程语言选择LAD（梯形图）。

然后，在FB编程区编写程序段1~19。

程序段1，M1.2为系统存储器始终为1。手/自动开关为1时，自动模式开启；手/自动开关为0时，手动模式开启。

程序段2，M1.2为系统存储器始终为1。启动使能按钮一个上升沿在步进电机未使能时，置位轴使能；启

动使能按钮一个上升沿在步进电机已使能时，复位轴使能。

程序段1、 2

程序段3，步进电机启动使能，调用MC\_Power指令程序。

程序段4，步进电机点动模式，调用MC\_MoveJog指令程序。

程序段3、 4

程序段5，在手动模式下，按复位按钮，复位指令启动。

程序段6，步进电机复位模式，调用MC\_Reset指令程序。

程序段7，步进自动转换为手动模式的一个脉冲时，触发暂停模式。

程序段5、 6、 7

程序段8，步进电机暂停模式，调用MC\_Halt指令程序。

程序段9，在自动模式下，按复位按钮，回原点指令启动。

程序段8、 9

程序段10，步进电机回原点模式，调用MC\_Home指令程序。

程序段11，在自动模式下，按自动启动按钮，开始执行定位运行。

程序段10、 11

程序段12，步进电机原点OK情况下，确认原点的偏差范围为-0.5~+0.5，则认为是原点位置。确认位置1

、2的定位偏差范围为-0.5~+0.5。

#### 程序段12

程序段13，自动开始启动按钮按下，如在原点位置，则调用15.0的定位速度和位置1的定位位置；如在位置1处，则调用20.0的定位速度和位置2的定位位置；

#### 程序段13

程序段14，定位1、2的触发，统一为定位触发。

程序段15，定位模式，通过上几步的设置为步进电机进行定位。

#### 程序段14、15

程序段16，自动指示灯程序。

程序段17，手动指示灯程序。

程序段18，步进电机启动使能程序

程序段19，步进电机使能异常报警程序。

#### 程序段16、17、18、19

后，我们要在主程序块下调用FB程序，直接在项目树中拖过来放开即可，这里要为FB的输入、输出分配好实参地址。

#### 主程序块调用FB

好了，西门子S7-1200PLC控制步进电机的部分梯形图程序就介绍到这里，实际工程应用大家自行增减。

## 西门子S7-1200PLC控制步进电机的SCL语言编程

### 函数块SCL语言程序

#轴使能启动(Axis:="步进电机",

Enable:=#轴使能\_1,

StartMode:=1,

StopMode:=1);

#复位启动(Axis:="步进电机",

Execute:=#手动模式\_1 AND #复位按钮);

"回原点启动"(Axis:="步进电机",

Execute:=#自动模式\_1 AND #复位按钮,

Position:=0,

Mode:=3);

#点动模式启动(Axis:="步进电机",

JogForward:=#正向点动 AND #手动模式\_1,

JogBackward:=#负向点动 AND #手动模式\_1);

#定位启动(Axis:="步进电机",

Execute:=#定位模式\_1,

Position:=#位置,

Velocity:=#速度);

#上升沿检测(CLK:=#手动模式\_1);

#暂停模式启动(Axis:="步进电机",

Execute:=#上升沿检测.Q);

//以上为指令块调用

#上升沿检测2(CLK:=#启动使能按钮);

IF #上升沿检测2.Q AND NOT #轴使能启动.Status THEN

    #轴使能\_1:= 1;

END\_IF;

IF #上升沿检测2.Q AND #轴使能启动.Status THEN

    #轴使能\_1:= 0;

END\_IF;

#自动模式\_1:= #"自/手动切换";

#手动模式\_1:= NOT #"自/手动切换";

//以上为按钮的状态

#步进启动2:= #自动模式\_1 AND #自动启动 AND "步进电机".StatusBits.HomingDone;

//以上定位方式的自动启动

IF #步进启动2 THEN

    CASE #程序步 OF

        0:

            #程序步:= 10;

        ;

        10:

#位置 := #位置1设置值;

#速度 := #速度1设置值;

#定位模式\_1 := 1;

#延时处理 := 0;

IF #定位启动.Done THEN

    #定位模式\_1 := 0;

#程序步:= 15;

END\_IF;

15:

  #IEC\_Timer\_0\_Instance(IN:=#延时处理,

    PT:=t#0s);

#延时处理 := 1;

IF #IEC\_Timer\_0\_Instance.Q THEN

    #位置 := #位置2设置值;

    #速度 := #速度2设置值;

    #定位模式\_1 := 1;

END\_IF;

    #延时处理 := 0;

#程序步:= 20;

END\_IF;

20:

  #IEC\_Timer\_0\_Instance\_1(IN:=#延时处理,

    PT:=t#10s);

#延时处理 := 1;

IF #IEC\_Timer\_0\_Instance\_1.Q THEN

#位置 := #位置3设置值;

#速度 := #速度3设置值;

END\_IF;

#程序步 := 10;

END\_IF;

END\_CASE;

ELSE

#位置 := 0;

#速度 := 0;

#自动模式\_1 := 0;

#程序步 := 0;

END\_IF;

//以上为三个位置的定位，预留延时功能

#自动指示灯\_1 := #自动模式\_1;

#手动指示灯\_1 := #手动模式\_1;

#报警指示灯\_1 := #轴使能启动.Error;

#轴使能指示灯 := #轴使能启动.Status;

#原点指示灯\_1 := "步进电机".StatusBits.HomingDone;

//以上为输出指示灯

Main程序块调用函数块