

```

;*****
;
;*本系统由 8155 负责拨码开关的读入,分为四位填充量(1%转单位)和一位速度选择档从 0-9
;*共十档速度,步进电机选用的是两相八拍制步距 0.9°的,调速控制由 8031 内部的 CT0 作为
;*定时器在两个脉冲间插入等待延时,赋以 CT0 的不同初值来调整延时时间,8031 的 CT0 以
;*所赋值开始向上计数,直到溢出为止,所以所赋值就应该是期望值相对于目标数的补数.
;*本系统实际真正用到的输入和输出有:
;* P1.4.....#PX!-OUT_IN 联接主轴凸轮的近接开关信号,作为填充料的主控制信号.从高电
;          平到低电平的跳变作为填充料的起始信号;
;* P1.0.....CP_OUT 为步进电机驱动器提供运行脉冲,每个负脉冲对应于步进电机的一步
;* P1.7.....ERR_OUT 错误报警指示灯输出,低电平有效(指示灯亮)
;* INT0.....ERROR-IN 驱动器出错时的反馈信号,指示驱动器有故障不能再发送脉冲.
;*所有其它控制脚全部没有用,都是多余的.
;*CT0 从 8155To 接入线无效,CT0 在定时器方式下时是以主频的 1/12 来进行计数定时的,按
;*CPU 的主频 6MHz 计算,CT0 就以 500KHz 来进行计数定时.每个计数单位对应于 2 微秒.
;*步进电机的升降速按每 11 步作为一个速度台阶,逐级进行的.所以,对应于每一档速度都控
;*制一个最小填充量,以满足升降速所占用的脉冲数(步进电机走步数),在每一档速度下,
;*对于不符合要求的填充量,均报警并死机,直到选择合理,并按下 RESET 钮后才能正常工作.
;*否则将拒绝工作.
;*****

```

```

                AJMP  START          ;初始化程序
                ORG   0003H
                LJMP  ERROR          ;INT0  错误报警处理(死机)
                ORG   000BH
                LJMP  INT_CT0        ;CT0  定时/计数器 溢出中断处理程序
                ORG   30H
START:          MOV   SP, #80         ;堆栈指针 50H,堆栈区 50H-7FH (留得比较大)
                MOV   TMOD, #17      ;0001 0001 设定 T0,T1 均为 16 进制计数的定时器
                MOV   IP, #5         ;0000 0101 置 T0,T1 均为优先中断级
                MOV   IE, #132       ;1000 0100 外部 INT1 允许,INT0 禁止
                MOV   TCON, #5        ;0000 0101 设 IT0,1 为电平触发方式
                MOV   2AH, #0        ;位地址 50H-57H,设定为两相四拍单向运转
MAIN1:          SETB  P1.7           ;P1.7 为报警信号输出,低电平有效
WAITOFF:       JNB   P1.4, $         ;等待主轴凸轮 OFF,加料控制的起始信号
WAITON:        JB    P1.4, $         ;等待主轴凸轮 ON
                LCALL READ_SPEED     ;读取调速参数(0-9)共 10 档
                LCALL READ_PULL      ;读取填充量参数,4 位十进制数
                LCALL PRO_PULL       ;将取入的填充量值转换为步数(扩大四倍)
                LCALL PRO_SPEED      ;调速参数的处理
                LCALL CTLOUT         ;物料填充控制输出(执行)
                SJMP  MAIN1          ;进入下一个重复过程(做下一包)

```

```

;*****
;* 填充量设定值读入程序.读入 4 位 10 进制数,经处理转换 16 进制数,数据储 *
;* 存在 CPU 内存的 4DH,4CH.所使用并影响的寄存器有:A,DPTR,R2,R3,R4,B *
;* 返回值:无. 4EH.4DH.4CH 作为以后输出时的依据 *
;* 操作对象: 8155,1248 拨码开关 *
;*****

```

```

READ_PULL:  MOV  A, #2           ;0000 0010 定义 PB 为输出方式,其余输入
            MOV  DPTR, #7FF8H ;000 指向命令状态口
            MOVX @DPTR, A
            MOV  DPTR, #7FFAH ;010 指向 PB 口(拨码开关的公共端)
            MOV  A, #254       ;1111 1110 PB 口输出值,最低位为 0
            MOVX @DPTR, A      ;输出.实际指向 4 位拨码开关的个位数.
            MOV  DPTR, #7FF9H ;001 指向 PA 口
            MOVX A, @DPTR      ;读取 4 位拨码开关的个位数
            ANL  A, #0FH       ;屏蔽高 4 位,仅取低 4 位
            MOV  4CH, A        ;暂存入 4CH 单元
            MOV  A, #253       ;1111 1101 指向拨码开关的十位数
            MOV  DPTR, #7FFAH ;010 指向 PB 口(拨码开关的公共端)
            MOVX @DPTR, A      ;使十位数拨码开关的公共端为 0
            MOV  DPTR, #7FF9H ;001 指向 PA 口
            MOVX A, @DPTR      ;读取 PA 口
            ANL  A, #0FH       ;屏蔽高 4 位,取出低 4 位,拨码开关的十位数
            SWAP A              ;高低 4 位对换.
            ORL  A, 4CH        ;将个位数(低 4 位)或入 A 累加器
            MOV  4CH, A        ;保存回 4CH 单元,至此个、十位数已读入 4CH
            MOV  DPTR, #7FFAH ;010 指向 PB 口(拨码开关的公共端)
            MOV  A, #251       ;1111 1011 指向拨码开关的百位数公共端
            MOVX @DPTR, A      ;百位数公共端置零电位
            MOV  DPTR, #7FF9H ;001 指向 PA 口
            MOVX A, @DPTR      ;读取百位数
            ANL  A, #0FH       ;屏蔽高 4 位,仅取低 4 位
            MOV  4DH, A        ;暂存 4DH 单元
            MOV  A, #247       ;1111 0111 指向拨码开关的千位数公共端
            MOV  DPTR, #7FFAH ;010 指向 PB 口(拨码开关的公共端)
            MOVX @DPTR, A      ;让千位数的公共端置零电位,准备读.
            MOV  DPTR, #7FF9H ;001 指向 PA 口
            MOVX A, @DPTR      ;读取千位数的值
            ANL  A, #0FH       ;屏蔽高 4 位,仅取低 4 位
            SWAP A              ;高低 4 位对调,即千位数移入高 4 位处
            ORL  A, 4DH        ;将百位数或入累加器 A
            MOV  4DH, A        ;合并后的数存入 4DH 单元
            MOV  R2, 4DH       ;重新取出刚才已读入的 4 位拨码开关数给
            MOV  R3, 4CH       ;操作寄存器 R2(高位),R3(低位)
            LCALL BCD_TO_HEX16 ;对携入 R2,R3 共 4 位的十进制数转换为 16 进制

```

```

MOV 4CH, R3 ;将已经转换为 16 进制的 16 位(二进制)数存入
MOV 4DH, R2 ;4DH(高位),4CH(低位)单元
RET ;本子程序结束返回
;
;*****
;* 调速设定值读入,查表译码程序.调速设定值保存在 32H,对应码表值存在 27H *
;* 影响寄存器:DPTR,A,硬件连接片 8155 *
;*****
READ_SPEED: MOV DPTR, #7FF8H ;指向 8155 命令状态口
MOV A, #1 ;指定 PA 口输出方式其它口为输入方式
MOVX @DPTR, A
INC DPTR ;指向 PB 口
INC DPTR ;调速拨码开关的公共端接地
MOVX A, @DPTR ;读取 PB 口数据
SWAP A ;高低 4 位交换,调速拨码开关连在 PB4-7
CPL A ;将累加器 A 按位取反,因硬件上是低有效
ANL A, #0FH ;保留低 4 位
MOV 32H, A ;保存在 32H 单元
MOV DPTR, #SPEED_TB1 ;指向调速码表
MOVC A, @A+DPTR ;按设定值对应查表
MOV 27H, A ;将查得值存入 27H
RET ;读数查表结束返回
;
;*****
;*调速码表.允许从 0-9 分十档调速 *
;*****
SPEED_TB1: DB 8H ;调速设定码 0,不升速,按第八级匀速运行, 72 转/分钟
DB 16H ;调速设定码 1,升速分 22-12=10 级 83 转/分钟
DB 27H ;调速设定码 2,升速分 39-12=27 级 207 转/分钟
DB 34H ;调速设定码 3,升速分 52-12=40 级 306 转/分钟
DB 43H ;调速设定码 4,升速分 67-12=55 级 393 转/分钟
DB 52H ;调速设定码 5,升速分 82-12=70 级 478 转/分钟
DB 60H ;调速设定码 6,升速分 96-12=84 级 556 转/分钟
DB 75H ;调速设定码 7,升速分 117-12=105 级 669 转/分钟
DB 7DH ;调速设定码 8,升速分 125-12=113 级 742 转/分钟
DB 7FH ;调速设定码 9,升速分 127-12=115 级 773 转/分钟
;
;*****
;*读入的填充量是百分之一转为单位的,换算电机每 400 步一转.即每 4 步对 *
;*应一个填充量单位.把读入数据乘以 4 然后放回 4DH(高位)4CH(低位), *
;*受影响寄存器 A,B,借用内存 30H. *
;*****
PRO_PULL: CLR C ;清进位标志
MOV A, 4CH ;取填充量低 8 位
MOV B, #4 ;乘数 4
MUL AB ;填充量乘 4,结果乘积 B(高 8 位)A 低 8 位
MOV 4CH, A ;保存填充量实际对应步数的低 8 位

```

```

MOV 30H, B ;暂存高 8 位
CLR C ;再清进位标志 C
MOV A, 4DH ;取填充量高 8 位
MOV B, #4 ;乘数 4
MUL AB ;填充量高位乘以 8
ADD A, 30H ;加上低 8 位乘积的高 8 位
MOV 4DH, A ;保存填充量实际对应步数的高 8 位
MOV 4EH, #0 ;再高 8 位清零
RET ;换算完毕,返回.

```

,* 对所设定的填充量和速度档位进行最小填充量判断, 较低的转速对应小的
,* 填充量,而较高转速就不能填充得太少否则还不够启动和停止升降速步数

```

PRO_SPEED: MOV A, 32H ;取出调速原始设定值
MOV B, #2 ;乘数 2(一个速度档转换为 2 字节数)
MUL AB ;速度乘 2
MOV DPTR, #PUL_MINI ;指向最小填充量表
MOVC A, @A+DPTR ;查表读取高 8 位
MOV R2, A ;暂转存 R2
INC DPTR ;指向下一位
MOV A, 32H ;再读回调速原始设定值
MOV B, #2 ;乘数 2
MUL AB
MOVC A, @A+DPTR ;查表读取高 8 位
MOV R3, A ;暂转存
MOV A, 4DH ;取填充量高 8 位
CLR C ;清进位标志
SUBB A, R2 ;设定填充量与最小填充量作高位比较
JC ERROR ;设定填充量小于最小填充量
JZ PS1 ;如果两者相等
RET ;如果调速参数小于走步数就不作处理返回.
PS1: MOV A, 4CH ;相等时就进一步比较低位,重取填充量的低 8 位
CLR C ;清进位标志
SUBB A, R3 ;设定填充量与最小填充量再作低位比较
JC ERROR ;如果设定填充量比最小填充量还小(A < R3),报警
RET ;校对合格 OK,返回

```

;不同转速下的最小填充量(每两字节为一档),高 8 位在前,低 8 位在后!

PUL_MINI:

DB	0H,	6EH,	1H,	76H,	2H,	94H,	3H,	9CH
DB	4H,	0BC H,	6H,	08H,	7H,	3CH,	9H,	06H
DB	9H,	0B8H,	0AH,	0EH				

,* 错误报警死机.错误种类有:速度选择错,驱动器反馈错误信号 *

```

ERROR:      CLR   EA           ;关中断.此后不接受任何中断请求
            CLR   P1.7        ;ERR 报警灯亮
HALT:       SJMP  $           ;死循环,死机

```

```

,*对携入 R2R3 10 进制数(4 位 10 进制数) 转换为 16 进制数 R2R3 出口
,*本程序调用子程序 BCD_TO_HEX8 进行一字节的转换,影响寄存器 A,B,R4

```

```

BCD_TO_HEX16:  MOV   A, R3      ;取 R3 低二位十进制数
               LCALL BCD_TO_HEX8 ;转成 16 进制
               MOV   R3, A      ;暂时放回 R3
               MOV   A, R2      ;取 R2 高二位十进制数
               LCALL BCD_TO_HEX8 ;转成 16 进制数
               MOV   B, #100    ;乘 100(百千位)
               MUL   AB
               ADD   A, R3      ;加上原低二位
               MOV   R3, A      ;存入 R3(16 进制的低二位)
               CLR   A          ;累加器 A 清零
               ADDC  A, B       ;带进位标志位加法,高二位 16 进制数
               MOV   R2, A      ;存入(返回)R2 带回
               RET              ;处理完毕,返回

```

```

,*2 位十进制数(BCD 码)转换为 16 进制数,调用输入 A...BCD 码,出口 A
,*影响寄存器:B,R4

```

```

BCD_TO_HEX8:  MOV   B, #16      ;除数 16(10H)
               DIV   AB         ;对原十进制数除 16
               MOV   R4, B      ;保留余数在 R4
               MOV   B, #10     ;乘数 10(十位)
               MUL   AB         ;对商乘 10(归回十位位)
               ADD   A, R4      ;加上余数
               RET              ;转换完成,返回

```

```

;*****
;控制输出程序.负责计数定时器的启动,对输出前的数据处理以及设置好控制所用*
;到的状态位,安排启动升速匀速运转以及降速停止三部分的脉冲数分配      *
;*****

```

```

CTL0UT:    MOV    26H, #11      ;26H 为 11 (0BH)每 11 步组成一个等速度小台阶
            MOV    A, 27H      ;取调速参数(已经转换过的)
            CLR    C          ;清进位标志
            SUBB   A, #13      ;A=A-13
            JC     CTL1       ;判断是否最低速档,是最低速档就转到 CTL1
            MOV    0BH, #12    ;设定第一工作区的通用寄存器 R3 初值,分级数
            INC    A          ;A=A+1
            MOV    27H, A      ;升速或减速级数

```

;从填充量中减去升降速段所走的步数,留下的是匀速段应该走的步数.每一个台阶约定 11 步
; 升降速要乘 2, 所以有运算 4EH.4DH.4CH - (27H-#12)*#22 => 4EH.4DH.4CH

```

            MOV    B, #22      ;每个台阶宽 11 步,来回共 22 步
            MUL   AB          ;A*B
            XCH   A, 4CH      ;Swap A & 4Ch
            SUBB  A, 4CH      ; |------(4CH)=(4CH)-A
            XCH   A, 4CH      ;Swap A & 4Ch
            MOV   A, 4DH      ;\
            SUBB  A, B        ;|--(4DH)=(4DH)-B
            MOV   4DH, A      ;/
            MOV   A, 4EH      ;取出最高位
            SUBB  A, #0       ;考虑借位
            MOV   4EH, A      ;存回原地址
            JC    CTL4       ;如果不够减就转 CTL4
            SJMP  CTL2       ;OK 转 CTL2
CTL1:    MOV    0BH, 27H     ;保存速度等级到第 1 工作区的 R3 中
            MOV    27H, #0   ;最低速档始终以第八档速度运行不再进行加减速
            MOV    R0, #76   ;R0 = 4CH
            LCALL  DEC_1     ;对已读入的 4EH.4DH.4CH 拨码开关设定数减一?
CTL2:    PUSH   PSW        ;保护程序状态寄存器 PSW
            MOV   PSW, #8   ;0000 1000 选择第一工作区
            MOV   R1, 27H   ;预置第一工作区的 R1,加速级数
            MOV   A, R3     ;按照表内的第 A 对数去
            LCALL CT0_REWRITE ;设置计数器 CT0 中的计数值
            MOV   R2, 26H   ;每级台阶宽度(11)
            CLR   0H        ;清状态位
            CLR   1CH      ;清状态位
            POP   PSW       ;恢复程序状态寄存器
            MOV   IE, #135  ;计数器 CT0 中断允许
            SETB  TR0       ;启动 CT0
CTL3:    JNB   1CH, $       ;等待 CT0 时间到溢出中断(中断处理发送脉冲)
            CLR   TR0       ;关 CT0

```

```

                MOV    IE, #132          ;关中断(禁止 CT0 溢出中断)
CTL4:  RET                                ;
;*****
;* 脉冲发出子程序, 一次调用发一个脉冲
;*****
RUN_STEP:  JB      51H,S1                ;三相六拍电机
           JB      50H,S2                ;反向运转
           CLR     P1.2                  ;CW-OUT 置 0,(正转)低电平有效
S4:        ACALL  S3                    ;短延时,以维持脉宽
           CLR     P1.0                  ;CP-OUT 置 0,运转脉冲发出
S6:        ACALL  S3                    ;短延时,以维持脉宽
           SETB   P1.0                  ;CP-OUT 置 1,关运转脉冲
           SETB   P1.1                  ;三相六拍电机保留位
           SETB   P1.2                  ;CW-OUT 置 1,关转向脉冲
           SETB   P1.3                  ;三相六拍电机保留位
           RET
S2:        SETB   P1.2                  ;CW-OUT 置 1,电机反向运转
           SJMP   S4
S1:        JB      50H,S5
           CLR     P1.3                  ; 三相六拍电机保留位
S7:        ACALL  S3                    ;短延时,以保持一定的脉宽
           CLR     P1.1                  ; 三相六拍电机保留位
           SJMP   S6
S5:        SETB   P1.3                  ; 三相六拍电机保留位
           JNC    S7
S3:        RET
;*****
;* 对由 R0 为起始地址的三字节数减 1,高位字节在后 *
;*****
DEC_1:  CLR     C                        ;清进位标志 (三字节数 4EH|4DH|4CH)
        MOV    A, @R0                    ;取出在 (4CH) 所存的数
        SUBB   A, #1                      ;减 1
        MOV    @R0, A
        INC    R0
        MOV    A, @R0                    ;(4DH)
        SUBB   A, #0
        MOV    @R0, A
        INC    R0
        MOV    A, @R0                    ;(4EH)
        SUBB   A, #0
        MOV    @R0, A
        RET

```

;启动参数表,步间的延时间隔,启动(停止)曲线数据,共 **122** 个 **16** 位数,低 **8** 位在前高 **8** 位在后.

WAIT_TBL:

DB	3BH,	0F6H,	0DCH,	0F7H,	5H,	0F9H,	0E4H,	0F9H
;1	5002	30	4168	36	3574	42	3128	48
DB	92H,	0FAH,	1DH,	0FBH,	8FH,	0FBH,	0EDH,	0FBH
;5	2780	54	2502	60	2274	66 零	2086	72
DB	3DH,	0FCH,	82H,	0FCH,	0BEH,	0FCH,	0F2H,	0FCH
;	1926	76 壹	1788	83	1668	89	1564	95
DB	20H,	0FDH,	49H,	0FDH,	6DH,	0FDH,	8EH,	0FDH
;9	1472	101	1390	107	1318	113	1252	119
DB	0ACH,	0FDH,	0C7H,	0FDH,	0E0H,	0FDH,	0F6H,	0FDH
;13	1192	125	1138	131	1088	137	1044	143
DB	0BH,	0FEH,	1EH,	0FEH,	30H,	0FEH,	41H,	0FEH
;17	1002	148	964	154	928	160	894	166
DB	50H,	0FEH,	5EH,	0FEH,	6CH,	0FEH,	78H,	0FEH
;21	864	172	836	178	808	184	784	189
DB	84H,	0FEH,	8FH,	0FEH,	9AH,	0FEH,	0A4H,	0FEH
;25	760	195	738	201 贰	716	207	696	213
DB	0ADH,	0FEH,	0B6H,	0FEH,	0BEH,	0FEH,	0C6H,	0FEH
;29	678	219	660	225	644	230	628	236
DB	0CEH,	0FEH,	0D5H,	0FEH,	0DCH,	0FEH,	0E3H,	0FEH
;33	612	242	598	247	584	253	570	259
DB	0E9H,	0FEH,	0EFH,	0FEH,	0FAH,	0FEH,	0FH,	0FFH
;37	558	265	546	271	524	282 叁	482	306
DB	13H,	0FFH,	18H,	0FFH,	1CH,	0FFH,	20H,	0FFH
;41	474	311	464	318	456	323	448	329
DB	24H,	0FFH,	27H,	0FFH,	2BH,	0FFH,	2FH,	0FFH
;45	440	335	434	339	426	346	418	352
DB	32H,	0FFH,	35H,	0FFH,	39H,	0FFH,	3CH,	0FFH
;49	412	357	406	362	398	369	392	375
DB	3FH,	0FFH,	42H,	0FFH,	45H,	0FFH,	47H,	0FFH
;53	386	381	380	387 肆	374	393	370	397
DB	4AH,	0FFH,	4CH,	0FFH,	4FH,	0FFH,	51H,	0FFH
;57	364	403	360	408	354	414	350	419
DB	54H,	0FFH,	56H,	0FFH,	58H,	0FFH,	5BH,	0FFH
;61	344	426	340	431	336	436	330	444
DB	5DH,	0FFH,	5FH,	0FFH,	61H,	0FFH,	63H,	0FFH

;65	326	449	322	455	318	460	314	466
DB	65H,	0FFH,	67H,	0FFH,	68H,	0FFH,	6AH,	0FFH
;69	310	472 伍	306	478	304	480	300	487
DB	6CH,	0FFH,	6EH,	0FFH,	6FH,	0FFH,	71H,	0FFH
;73	296	493	292	500	290	503	286	510
DB	73H,	0FFH,	74H,	0FFH,	76H,	0FFH,	77H,	0FFH
;77	282	517	280	521	276	528	274	532
DB	79H,	0FFH,	7AH,	0FFH,	7BH,	0FFH,	7DH,	0FFH
;81	270	540	268	543	266	547 陆	262	556
DB	7EH,	0FFH,	7FH,	0FFH,	81H,	0FFH,	82H,	0FFH
;85	260	560	258	564	254	572	252	577
DB	83H,	0FFH,	84H,	0FFH,	86H,	0FFH,	87H,	0FFH
;89	250	581	248	586	244	595	242	600
DB	88H,	0FFH,	89H,	0FFH,	8AH,	0FFH,	8BH,	0FFH
;93	240	605	238	610	236	615	234	620
DB	8CH,	0FFH,	8DH,	0FFH,	8EH,	0FFH,	8FH,	0FFH
;97	232	625	230	630	228	635	226	641
DB	90H,	0FFH,	91H,	0FFH,	92H,	0FFH,	93H,	0FFH
;101	224	646	222	652	220	658	218	663
DB	94H,	0FFH,	95H,	0FFH,	96H,	0FFH,	97H,	0FFH
;105 柒	216	669	214	675	212	681	210	688
DB	98H,	0FFH,	99H,	0FFH,	9BH,	0FFH,	9DH,	0FFH
;109	208	694	206	701	202	714	198	728
DB	9FH,	0FFH,	0A1H,	0FFH,	0A3H,	0FFH,	0A5H,	0FFH
;113 捌	194	742	190	757 玖	186	773	182	789
DB	0A7H,	0FFH,	0A9H,	0FFH,	0ABH,	0FFH,	0ADH,	0FFH
;117	178	806	174	824	170	842	166	862
DB	0AEH,	0FFH,	0AFH,	0FFH				
;121	164	872	162	882 微秒单位	转/分钟			

;按此表换算步进电机的最低转速为 30 转/分钟,最高转速约 882 转/分钟,按 170 微秒的倒数
;除以 400,在乘上 60 秒,等于 882 转/分钟,本程序实际只编到第 115 这一级,换算得本机的实际
;最高转速为:773 转/分钟.对应与 0 档速度选择时的实际转速为 72 转/分钟.

```

;*****
;
;*设置 CT0 计数器数据,数据来源为上表 WAIT_TBL,取哪一对数由携入 A      *
;*表内偏移量指定                                                         *
;*****
CT0_REWRITE:MOV  DPTR, #WAIT_TBL      ;启动曲线表
              RL   A                  ;左移一位,A=A*2 指向 16 位表格
              PUSH ACC                ;保护累加器 A
              MOVC A, @A+DPTR        ;取一字节输出
              MOV  TL0, A             ;T0 低八位
              INC  DPTR               ;指向下一位
              POP  ACC                ;弹出累加器 A
              MOVC A, @A+DPTR
              MOV  TH0, A            ;T0 高八位
              RET

```

```

;*****
;
; *CTO 计数定时器溢出中断处理程序 *
; *启用内存位 0H 作为升降速标志: 0...升速,当 R1 亦为零时为匀速状态 *
; * 1CH 作为程序处理结束与否的出口标志, 0...正在处理中; 1...处理结束 *
;通用寄存器的分工: R1.....升速或降速的剩余级数,每升或降一级速时均减小 1 *
; * R2.....同一个速度台阶还剩下的步数,约定宽度为 11 *
; * R3.....当前速度级,对应于 130 延时量中的寻址指针(乘 2 后) *
;*****

```

```

INT_CT0:   PUSH  PSW           ;保护程序状态寄存器
           PUSH  ACC           ;保护累加器 A
           MOV   PSW,#8        ;设置程序状态寄存器,选择第一工作区
           MOV   A,R3          ;延时量表内偏移量(动态变化的)
           ACALL CT0_REWRITE   ;重新设定 CT0 计数值
           JNB  0H,K1          ;0H 位单元作为升降速标志,0 代表升速,1 减速
;*****

```

```

; * 减速度段,以每级 11 步组成一个等速小台阶,逐级减速直到 R1 减到零,再转 K2 停止 *
;*****

```

```

           MOV   A,R1          ;R1 减速级数,是一个无符号数
           JZ    K2            ;R1=0 说明现在减速已经完成,转 K2 立即停止!
           ACALL RUN_STEP      ;脉冲输出 (一个脉冲)
           DJNZ  R2,K3         ;本级速度台阶宽度到了没有,还没有到,返回
           MOV   R2,26H        ;26H 约定数为 11,修改它就可以修改台阶宽度
           DEC   R3            ;减一级速,下一个台阶
           MOV   A,R3          ;
           JNB  ACC.7,K4       ;是否已输出完,输出直到 R3 等于零以后(负)
           MOV   R3,#0         ;清 R3=0
K4:        DJNZ  R1,K3         ;R1
K2:        CLR   TR0          ;关计数器 CT0
           CLR   ET0          ;关计数器中断
           SETB 1CH           ;1CH 已完毕标志,指示主程序此后可以跳出循环
K3:        POP   ACC          ;恢复累加器
           POP   PSW          ;恢复程序状态标志
           RETI                ;计数器中断返回
;*****

```

```

; * 升速度段,以每级 11 步组成一个等速小台阶,逐级上升直到 R1 减到零,再转匀速段 *
;*****

```

```

K1:        MOV   A,R1          ;升速剩余级数,升速还有多少级没有完成?
           JZ    K5            ;已经完成了,就转匀速段
           ACALL RUN_STEP      ;输出一个脉冲. 继续处在升速阶段
           DJNZ  R2,K3         ;本级速度台阶宽度到了没有,还没有到,返回
           MOV   R2,26H        ;台阶宽度到,准备下一个台阶宽度(#11)
           INC   R3            ;速度升一级
           DJNZ  R1,K3         ;R1(速度位置指针)减 1 后不为零就返回,尚未升完速

```

```

DEC R3 ;升速已经完成,减去多升的那级,实际已进入匀速段
MOV R0,#76 ;4CH 匀速填充量数据(低位)起始地址
ACALL DEC_1 ;本脉冲计算在匀速段,从该段剩余填充量中补减一
AJMP K3 ;返回

```

;* 匀 速段,当判断匀速段计数减到零以后就设置状态位 0H 为 1,以示匀速段已经结束*
;*下一次进入时就转作减速段输出,以逐步减速准备停止 *

```

K5: ACALL RUN_STEP ;脉冲输出 (一个脉冲)
MOV R0,#76 ;4CH 匀速填充量数据(低位)起始地址
ACALL DEC_1 ;对填充量减一
JNC K3 ;还够减返回,继续匀速段
MOV R1,27H ;匀速段输出结束,重取升速级数,以作减速级数用
SETB 0H ;置升降标志位为 1,以指示下一步应该开始降速
AJMP K3 ;完成一次中断服务,返回
END ;整个控制程序到此结束

```

;本程序在实际使用中尚存在如下问题有待进一步改善:

; 1.受干扰情况时有发生,会产生一包料填充两次的情况,估计是主轴近接开关受干扰,在一次填充完成以后有干扰串入,程序当作是充填指令执行了;插入一个短小延时程序,再读近接开关状态,用脉宽来确认近接开关信号的有效性,屏蔽干扰信号,避免重复充填.

; 2.填充料的计量精度不够,主要是物料有时会粘留在螺杆头上,而有时又会成块的跌落,造成填充物料时多时少,当包装填充量较小的时候尤为明显,对毫克单位的填充根本无法实现,这与产品说明书上所说的计量包装范围有较大的出入.解决办法设想:

; a.对与要求用于小包装的用户,另外配一套小螺杆,以适应毫克级的计量需要;

; b.在一次充填的过程的结尾停转后,加入反转回抽动作,让物料有意识的退回一点,让该落下的下落,不该下落的留在螺杆内,防止误落料,减少计量误差.

; 3.从本系统的功能上来说,考虑作为粉剂机的专用控制单元,不必留有那么多的余地,所谓专机专用,电路尽量简单,可以考虑不用 8155 接口片子,CPU 可以改用 89C51,程序做在 CPU 片子内,有利于加密,减少外围电路.让 CPU 的端口直接用来读取拨码开关数据.

; 4.根据最高转速推算,包装机的最低包装速度是每分钟 30 包,加上预留部分间隔时间和启动停止时的升降速,步进电机的转数最多不到 10 转.所以设计的四位拨码开关中的最高位是永远都不可能用到的.也就是多余的浪费,应该把它省掉.

; 根据以上几点修改,控制程序亦要作出适当修改.

;

1999.1.17

以上是我对一个步进电机控制器产品 eprom 程序的反汇编解读纪录,边读边整理,可能有些地方理解不够确切,这是我学习单片机控制的入门的第一单任务,不久这个产品目前被淘汰.今天拿出来供各位从事单片及工业控制的朋友们参考.

晓奇 11.27.2002