江 蘇 大 學

JIANGSU UNIVERSITY

微机接口课程设计报告

步进电机综合控制

设计时间：2019 — 2020 学年 第 1 学期

专业班级：不告诉你

学生姓名：nosteglic

学　　号：不告诉你

指导教师：不告诉你

开课系部：计算机科学与技术系

2020 年 1 月 5 日

目录

步进电机综合控制

一、 课程设计目的

1、熟悉各常用可编程接口芯片的工作原理、使用方法及引脚连接特性，正确操作实验设备，能根据设计要求制定总体方案，完成硬件连线，承担责任。

2、能够根据要求编写出满足功能要求的软件代码，能够利用开发工具进行设计调试、协助分析运行结果、定位设计错误等。

3、具有创新意识。尝试对所要求的功能设计提出有效的改进设想并努力实现；或尝试增加新的系统功能并努力实现。

4、具有对微机输入输出系统进行需求分析的能力，能针对具体的问题获取新知识完成系统的设计。

5、能够以口头和书面方式准确地描述、总结所完成的设计和主要成果，撰写比较规范的课程设计报告。

6、掌握综合控制键盘、数码管、步进电机的能力。

7、掌握有关中断服务程序的编制方法。

二、 设计内容及具体要求

2.1 设计内容

1、通过按键实现步进电机的“启动/停止”、“正反转”、“设定转速”、“设定步数”、“程序停止”等功能。同时，将电机的当前转速和步数显示在数码管上。

2、在基础课程设计的前提下，进行创新。

2.2 具体要求

1、键盘命令如下：

表 1 键盘命令

命令	功能
A	启停步进电机
B##	步进电机以速度##转/分正转（顺时针）
C##	步进电机以速度##转/分反转（逆时针）
D##	步进电机前进##步（顺时针）
E##	步进电机后退##步（逆时针）
F	程序结束

2、用左边 3 个数码管显示命令（B、C、D、E）和当前设定的转速/步数。

3、用右边 5 个 LED 数码管显示当前的结果。假设当前步数初值为 0：

（1）每正转一步，步数值加 1；

（2）反转一步，步数值减 1。

用右边 5 个 LED 数码管显示当前的结果。具体步进显示格式如下：

表 2 步进显示格式

显示	功能
0.####	表示步数大于 0
--.####	表示步数小于 0

4、8251、8253、8255、8259、0832、0809 等接口芯片模块必须用到 3 个或 3 个以上。

5、必须用到键盘和 LED 数码管模块进行相关的设置和显示。

三、 所需实验器件

3.1 B3 区：8259 电路

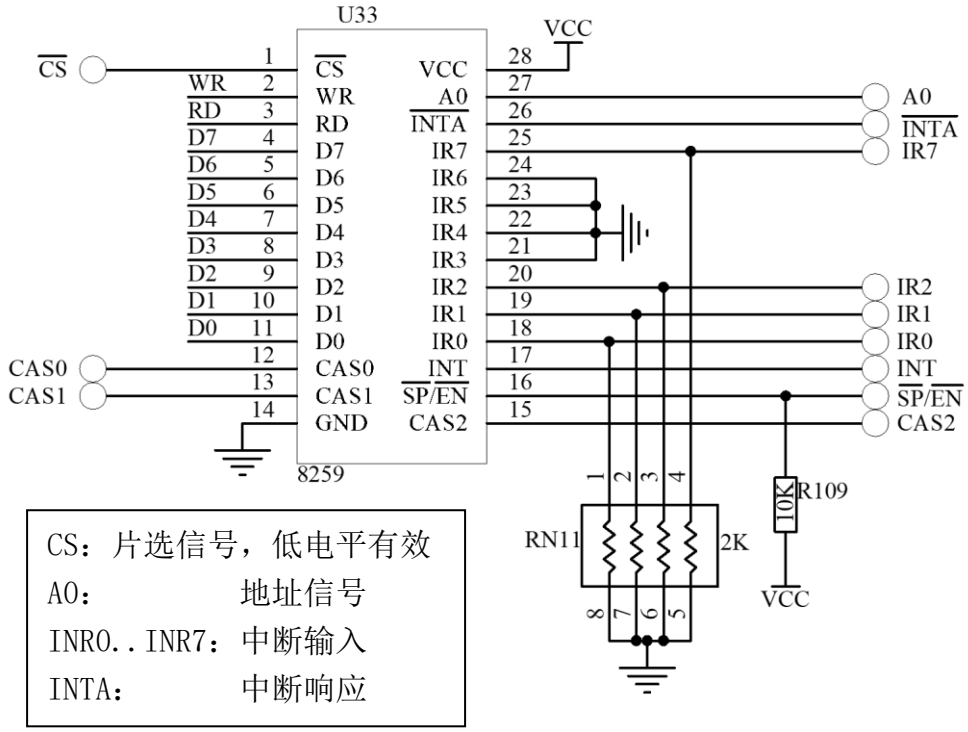


图 1 8259A——B3 区

通过 8259 处理中断。8259 向 CPU 提供中断类型号，执行相应的中断服务程序：使步进电机转动一步——通过切换每相线圈的电流顺序实现电机的步进式旋转。

3.2 D3 区：8255 电路、数码管驱动电路

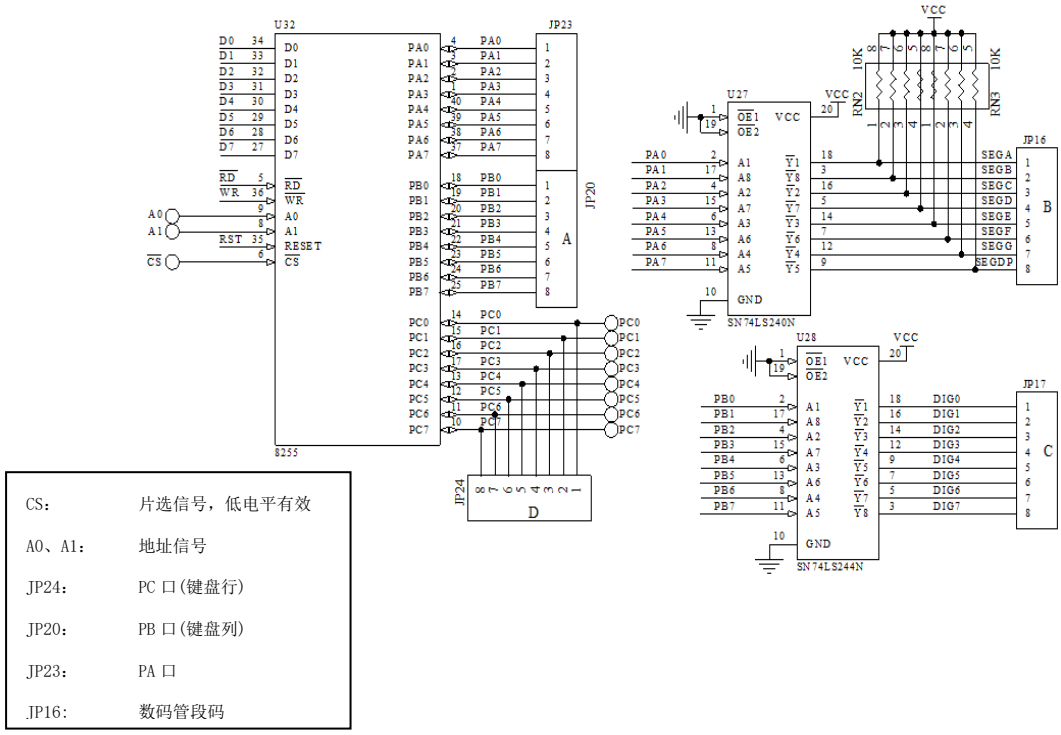


图 2 8255A——D3 区

1、通过 8255 的 PC4~PC7 口控制步进电机的驱动。

2、通过 8255 系列模块控制键盘的输入和数码管的显示。

3.3 C4 区：8253 电路

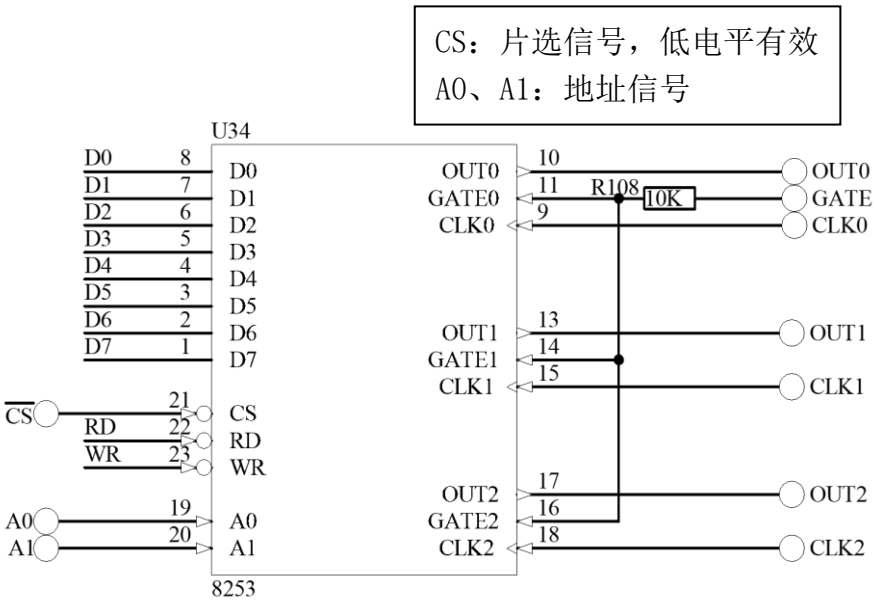
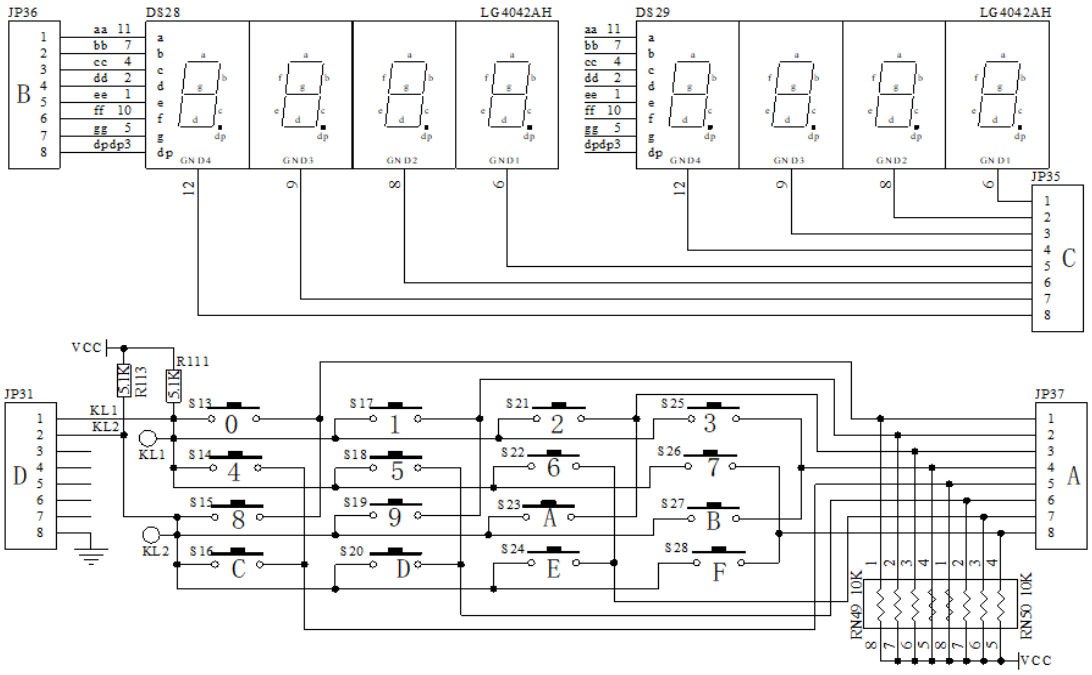


图 3 8253——C3 区

通过 8253 产生步进电机驱动的定时信号，即产生中断的频率。

3.4 F5 区：键盘&LED



A：	按键的列线	B：	数码管段码
C：	数码管选择脚	D：	按键的行线

图 4 键盘和数码管——F5 区

用于输入功能键和设置的步数/转速，并显示设置的功能、功能参数和当前步数值。

3.5 D1 区：步进电机

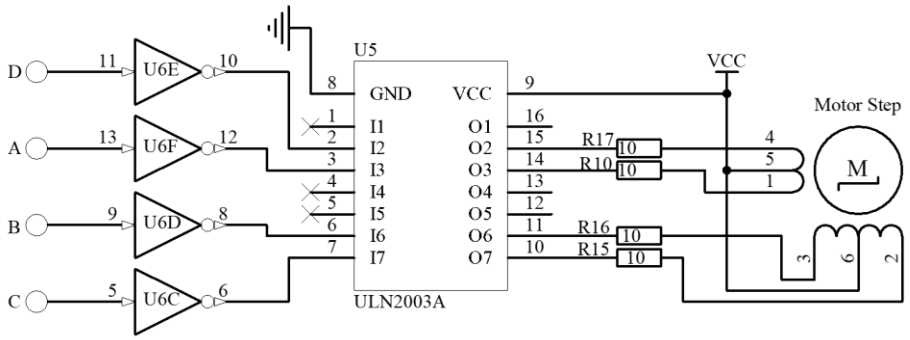


图 5 步进电机——D1 区

设计核心模块。

3.6 F4 区：发光管、开关

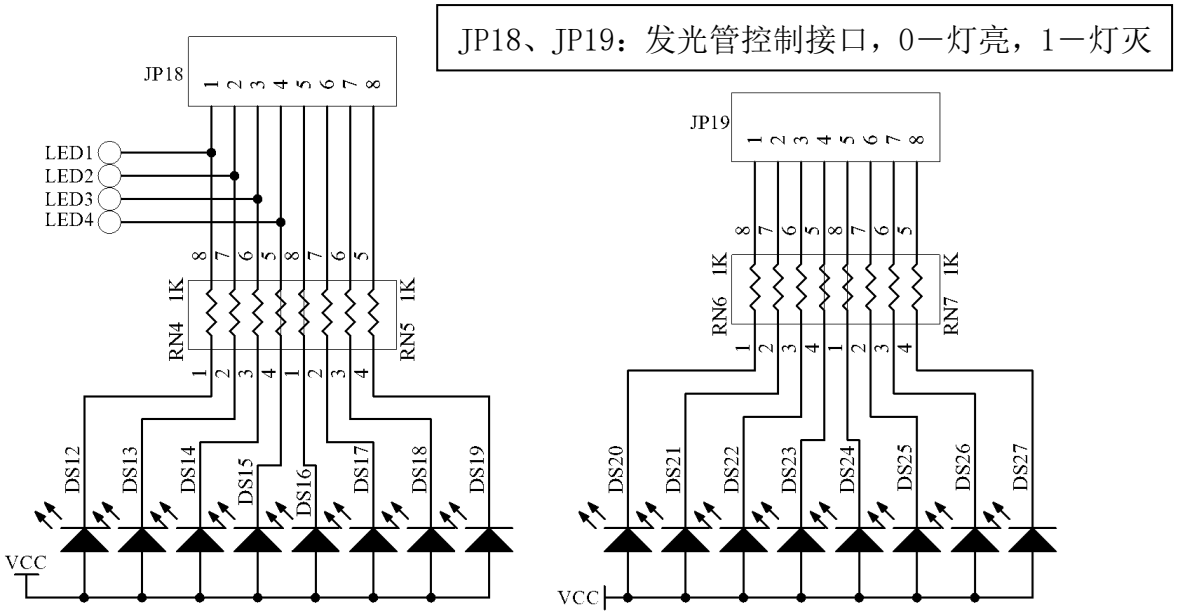


图 6 发光管——F4 区

发光管用于观察 8251 发送器准备好信号 TxRDY 和接收器准备好信号 RxRDY 的状态。

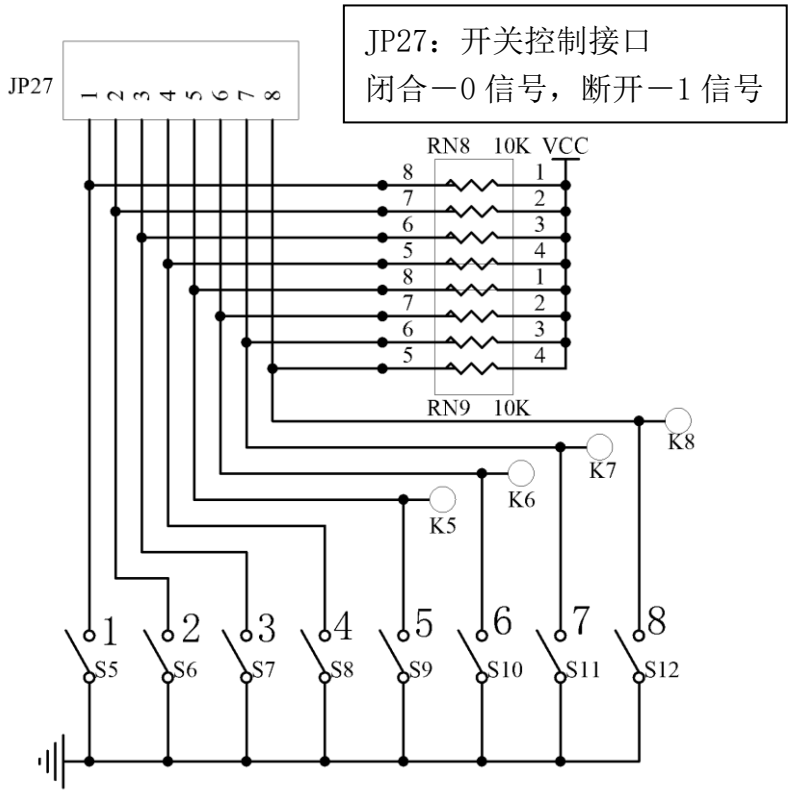


图 7 开关——F4 区

开关用于设置步进电机的转动方式。

3.7 C3 区：8251 电路

CS：	片选信号，低电平有效；	RxC、TxC： 收发时钟
C/D：	命令/数据；	RXD、TXD： 串行收发
CLK：	时钟	

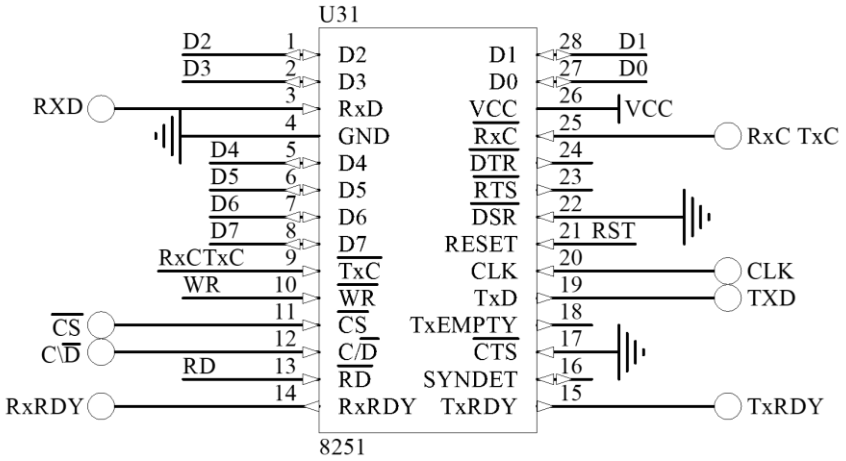
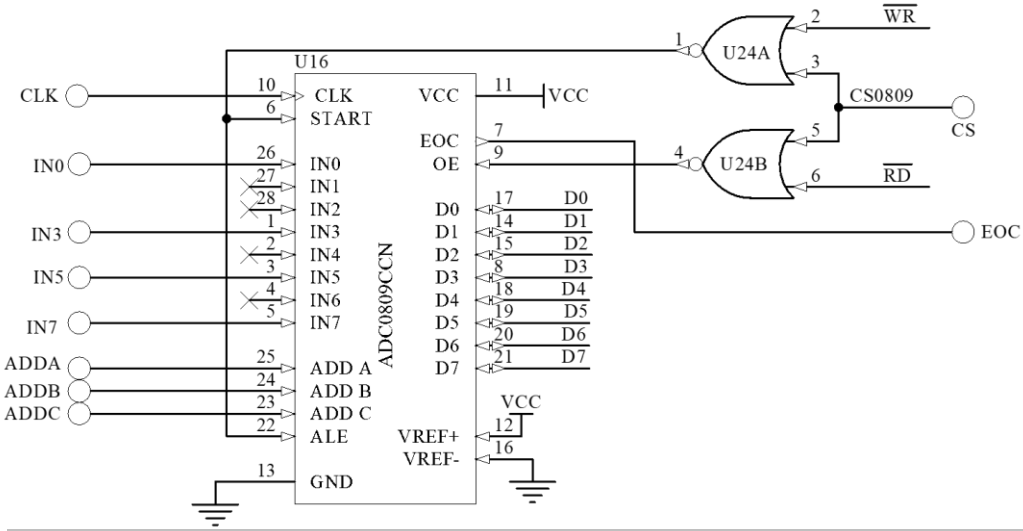


图 8 8251——C3 区

8251 用于控制两台步进电机之间的通信。

3.8 C2 区：ADC0809 模数转换



CS：	片选，低电平有效；
CLK：	输入时钟(10k－1280kHZ)；
ADDA，ADDB,ADDC：	通道地址输入口；
EOC：	转换结束标志，高电平有效。
IN0、IN3、IN5、IN7：	模拟量输入

图 9 ADC0809——C2 区

0809 预想用于接收机的启停。

四、 设计的方案及电路原理图

4.1 设计方案

1、查阅资料，了解步进电机的转动原理以及各芯片的工作原理。

2、采用模块化的设计方法，减轻主程序负担，便于调试和修改。

3、进行分工合作，最后整合调试。

4、在基础实验成功的前提下，进行适当的改进和创新。

我们通过按键控制给出步进电机需要完成的功能及相应的功能参数，并将需要设置参数的功能键值（B、C、D、E）、设置的步数/转速和当前转动所产生的步数值通过数码管显示出来。

特殊化负责启停的功能键 A 和负责终止程序的功能键 F。任何情况下，按下 F 都将终止程序运行，即停机。只有当电机停止转动时才能设置相应的功能，否则键入功能键和相应参数时，数码管显示将不予修改且电机状态不予改变。电机转动时，只有按下 A 才能使电机停止转动，并且只能通过 A 启动步进电机。

通过开关量控制可设置步进电机的转动方式：单 4 拍、双 4 拍和单双 8 拍。

以下分成端口分工、硬件连线、变量设置、程序模块组成等部分进行介绍。

4.1.1 使用端口分工

表 3 使用端口的分工

模块	硬件端口	端口地址	端口名称	端口作用
8255 [CS1]	A 口	270H	PORT8255_A	数码管段码
	B 口	271H	PORT8255_B	位码（数码管选择脚） 控制按键的列线
	C 口	272H	PORT8255_C	PC0、PC1→控制按键的行线 PC4~PC7→分别控制电机 ABCD 四相位
	控制端口	273H	PORT8255_K	写方式控制字
8259 [CS3]	偶地址	250H	PORT8259_0	写 ICW1、OCW2
	奇地址	251H	PORT8259_1	写 ICW2、ICW4、OCW1
8253 [CS2]	计数器 0	260H	PORT8253_0	分频，用于产生中断的频率
	控制端口	263H	PORT8253_K	写方式控制字，计数初值
8251 [CS4]	偶地址	240H	PORT8251_0	写模式字、控制字
	奇地址	241H	PORT8251_1	用来接收或发送数据
0809 [CS5]	IN0	230H	PORT0809_0	模数转换

4.1.2 硬件连线

表 4 硬件连线

模块	区域	端口	连接	区域	端口
8255	D3 区	8255 片选 CS*、A0、A1	连接	A3 区	系统 CS1*、A0 、A1
		PC0、PC1	连接	F5 区	KL1、KL2
		JP20(PB0-PB7)	连接	F5 区	A=JP37(1-8)（键盘列线）
		(A 口)B=JP16(SEGA-SEGP)	连接	F5 区	B（JP36）（段码）
		(B 口)C=JP17(DIG0-DIG7)	连接	F5 区	C（JP35）（位码）
		PC2、PC3	连接	F4 区	K5、K6
步进电机	D1 区	A	连接	D3 区	PC4
		B	连接	D3 区	PC5
		C	连接	D3 区	PC6
		D	连接	D3 区	PC7
8259	B3 区	8259 片选 CS*、A0	连接	A3 区	系统 CS3*、A0
		INT、INTA*	连接	A3 区	INTR、INTA*
		IR0	连接	C4 区	OUT0
8253	C4 区	8253 片选 CS*、A0、A1	连接	A3 区	系统 CS2*、A0 、A1
		CLK0	连接	B2 区	1M
		GATE	连接	C1 区	VCC
8251	C3 区	8251 片选 CS*、C/D	连接	A3 区	系统 CS4*、A0
		RxC、TxC	连接	B2 区	62.5K
		TXD	连接	C3 区	RXD
		TxRDY、RxRDY	连接	F4 区	LED1、LED2
		CLK	连接	B2 区	4M
预实现					
0809	C2 区	0809 片选 CS*	连接	A3 区	CS5*
		ADDA、ADDB、ADDC	连接	A3 区	A0、A1、A2
		CLK	连接	B2 区	500K
		IN0	连接	F6 区	0~5V
蜂鸣器	F8 区	Ctrl	连接	D3 区	接收机 PC0

4.1.3 变量设置

为了各模块的衔接完整和程序的正常运行，考虑设置以下变量：

表 5 变量设置

变量名	类型	说明
BITCODE	DB	存放数码管位码值
SEGTAB	DB	存放段码值 0~F、“0.”、“-.”
KEYVALUE	DB	存放键值 0~F、“0.”、“-.”
KEYCODE	DB	存放扫描按键得到的行列码
buffer	DB	显示缓冲区
bFirst	DB	停机标志　　0：启动中　1：停止
bClockwise	DB	转动方向标志　0：逆时针　1：顺时针
bNeedDisplay	DB	数码管更新显示标志
StepControl	DB	下一次要送给步进电机的相位值
SetCount	DB	##
StepCount	DB	当前步数绝对值
StepDec	DW	通过功能键 D、E 设置的步数
SpeedNumber	DB	通过功能键 B、C 设置的转速级数
JiShu	DB	通过转速级数计算得到的 8253 计数初值
StyleMove	DB	转动方式选择　0：单 4 拍　1：双 4 拍　2：单双 4 拍
ControlStyle	DB	相位值表，用于相位赋值
DONE	DB	中断完成标志　0：未完成　1：完成
		预设计
STOPPINGTEST	DB	停机标志（调节旋钮 0FFH）

4.1.4 程序模块组成

具体在软件的流程图中介绍。

1、主程序

（1）判断有无按键按下；

（2）判断数码管是否需要更新显示；

（3）判断电机状态：启动/停止；

（4）判断设定的步数是否已走完；

（5）判断按键功能和设置步数/转速。

2、初始化模块

（1）【INIT8253】8253 初始化：进行相应转速的设置，对 1M 进行分频；

（2）【INIT8255】8255 初始化：进行电机的步进控制和键盘解析；

（3）【INIT8259】8259 初始化：处理中断；

（4）【INIMOTOR】步进电机初始化：设置停机标志、转动方式、显示缓冲区；

（5）【INTERRUPT_VECTOR】中断向量表初始化：设置向量表偏移地址和段基地址，保护断点。

（6）【INIT8251】8151 初始化：设置接收机和发送机通信。

3、子程序

（1）【LED_DISPLAY】数码管显示：将 buffer 中的数据通过数码管显示出来；

（2）【SCANKEY】扫描按键：扫描键盘得到按键的行列码存入 AL；

（3）【TRANSLATE】行列码转换为键值：将扫描得到的行列码转换成相应的键值；

（4）【GetBuffer65】设置步数/转速：得到##值送入 SetCount，以便设置步数/转速；

（5）【SetJiShu】设置 8253 计数值：将##（SetCount）送入 SpeedNumber（转速级数），再根据转速级数通过公式$5000 + 100 \times SpeedNumber$得到 JiShu（计数值）。

（6）【HltFunction】功能 F 显示：按 F 显示全“F”表示停机；

（7）【StepCountSet】得到当前步数值：将显示缓冲区后 4 位转换成 4 位数存入 StepCount（当前步数值）；

（8）延时【DELAY】系列：进行一定时间的延时；

（9）【AlterStep】步进电机步数调整。

（10）【StepControlSet】设置步进电机下一步相位值。

（11）【SetStyleMove】设置步进电机的转动方式。

（12）【SEND8251】发送机发送数据给接收机。

（13）【RECEIVE8251】接收机接收发送机发送过来的数据。

4、中断子程序 TIMERO

（1）传送相位值给步进电机控制其旋转一步，并调整下一步要给出的相位值；

（2）判断通过 D、E 功能设置的步数是否已走完。

4.2 电路原理图

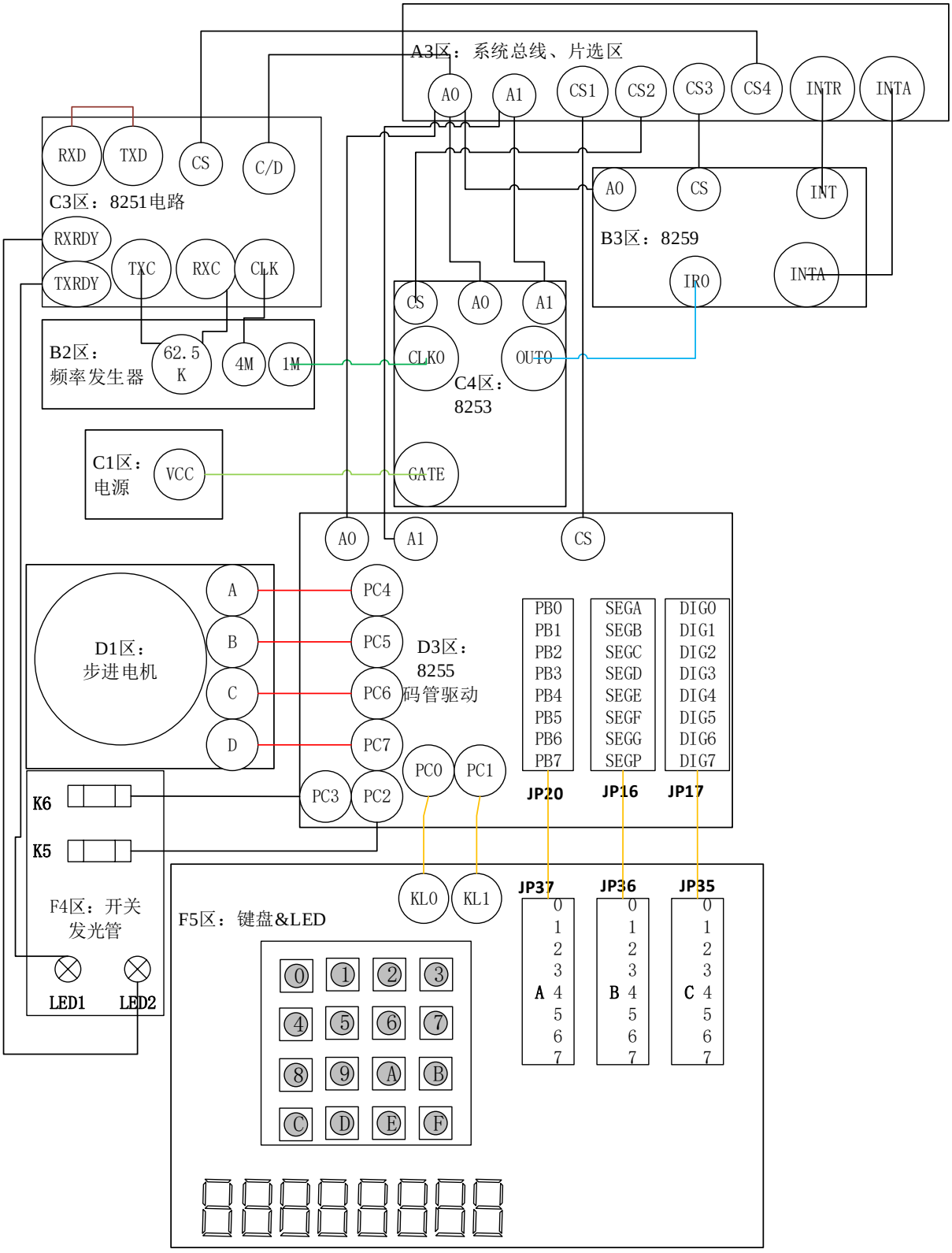


图 10 电路原理图

五、 软件的流程图

本设计由小组成员分工完成，每个人着重介绍自己设计（即打*）的模块。

5.1 *主程序

首先调用初始化模块对相应的模块进行初始化，并调用 SetStyleMove 子程序设置步进电机的转动方式。只有设置好了转动方式，步进电机才可以正常运行。

调用 LED_DISPLAY 数码管显示子程序，将电机的初始状态显示出来。接着调用 SCANKEY 子程序进行按键扫描，若没有按键按下，需要根据数码管更新标志 bNeedDisplay 标志判断数码管是否需要更新显示，若其为 0，则维持现状不变，继续扫描按键，否则需要通过停机标志 bFirst 判断电机是否处于启动状态：

（1）若 bFirst 为 1，则代表电机应停止转动，关闭中断 CLI；

（2）反之，电机应开中断 STI，进行转动。

若由按键按下，由于功能键 A 控制电机的启动和停止，所以先判断是否键入 A：

（1）若键入 A，则将当前启动的电机停止或将当前停止的电机启动，若为前者（bFirst 由 0 变 1），则应关闭中断，停止电机转动；否则应判断此时应处于哪种功能模式下：

1° 若处于 BC 设置转速模式下，则代表电机没有指定步数限制（置 StepDec 为-1），除非按 A 停机或按 F 结束程序，否则电机将以设定的速度维持转动；

2° 若处于 DE 设置步数模式下，则代表电机受设置的步数限制，当运行指定步数后电机将自动停止转动。而在 DE 模式下的运行转速，应为最近一次通过 BC 设置的速度为准。由于电机转速比较快，1~99 步的设置相比于电机速度来说意义不大，因此我们设定，每十步为 1 个单位，例如输入 10，则代表电机应转 100 步。

（2）若没有键入 A，输入了其他功能键值，则判断是否是功能 F 结束整个程序，如若不是，则需要根据停机标志 bFirst 判断此时电机所处的运行状态。我们设置，只有功能键 A 可以控制电机的启停，因此当电机正在转动时，输入的功能键 B、C、D、E 和功能参数应无效，即不改变电机当前状态，也不显示新功能。只有当电机停机但未结束程序时可设置电机的功能。

（3）当设置电机功能时，转速和步数只能设置数字，因此将与字母功能键区分开来，即输入数字只改变电机的转速和所走步数，输入字母只改变电机的功能。如果键入的是 B 和 D，则设置电机为正转，如果键入的是 C、E，则设置电机为反转，通过改变 bClockwise 标志来改变电机的转动方向。

（4）设置完功能和相应参数后，按 A 启动电机。

（5）键入 F，整个程序将停止运转，此后所有按键无效。

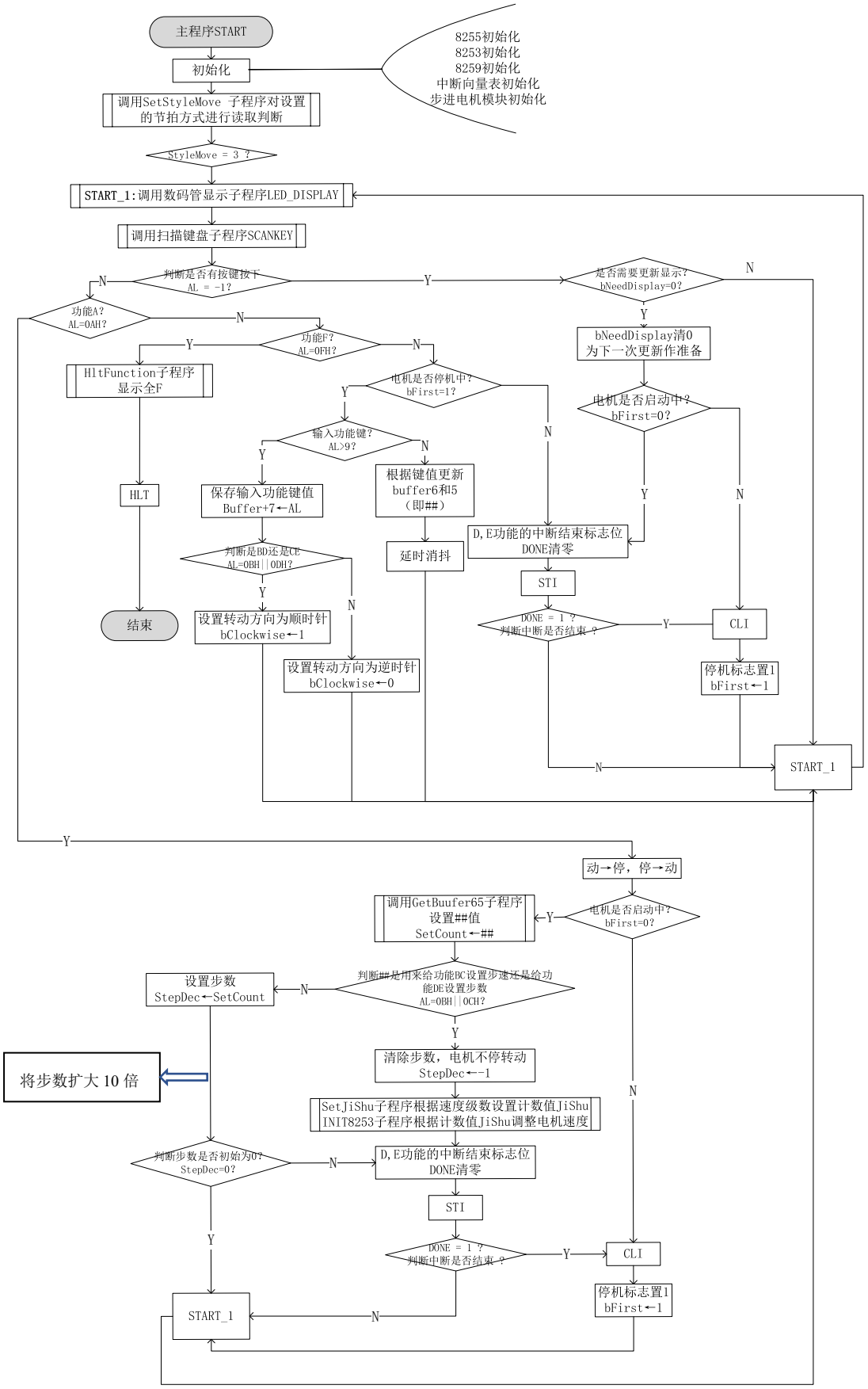

主程序START
初始化
8255初始化
8253初始化
8259初始化
中断向量表初始化
步进电机模块初始化
调用SetStyleMove 子程序对设置的节拍方式进行读取判断
StyleMove = 3 ?
START_1:调用数码管显示子程序LED_DISPLAY
调用扫描键盘子程序SCANKEY
判断是否有按键按下 AL = -1?
是否需要更新显示? bNeedDisplay=0?
功能A? AL=0AH?
功能F? AL=0FH?
bNeedDisplay清0 为下一次更新作准备
HltFunction子程序 显示全F
电机是否停机中? bFirst=1?
电机是否启动中? bFirst=0?
输入功能键? AL>9?
HLT
保存输入功能键值 Buffer+7←AL
根据键值更新 buffer6和5 (即##)
D,E功能的中断结束标志位 DONE清零
判断是BD还是CE AL=0BH||0DH?
延时消抖
STI
结束
设置转动方向为顺时针 bClockwise←1
设置转动方向为逆时针 bClockwise←0
DONE = 1 ? 判断中断是否结束 ?
CLI
停机标志置1 bFirst←1
START_1
动→停，停→动
调用GetBuufer65子程序 设置##值 SetCount←##
电机是否启动中? bFirst=0?
判断##是用来给功能BC设置步速还是给功能DE设置步数 AL=0BH||0CH?
设置步数 StepDec←SetCount
将步数扩大 10 倍
清除步数，电机不停转动 StepDec←-1
SetJiShu子程序根据速度级数设置计数值JiShu
INIT8253子程序根据计数值JiShu调整电机速度
判断步数是否初始为0? StepDec=0?
D,E功能的中断结束标志位 DONE清零
STI
DONE = 1 ? 判断中断是否结束 ?
CLI
停机标志置1 bFirst←1
START_1
Y
N

图 11 主程序流程图（竖）

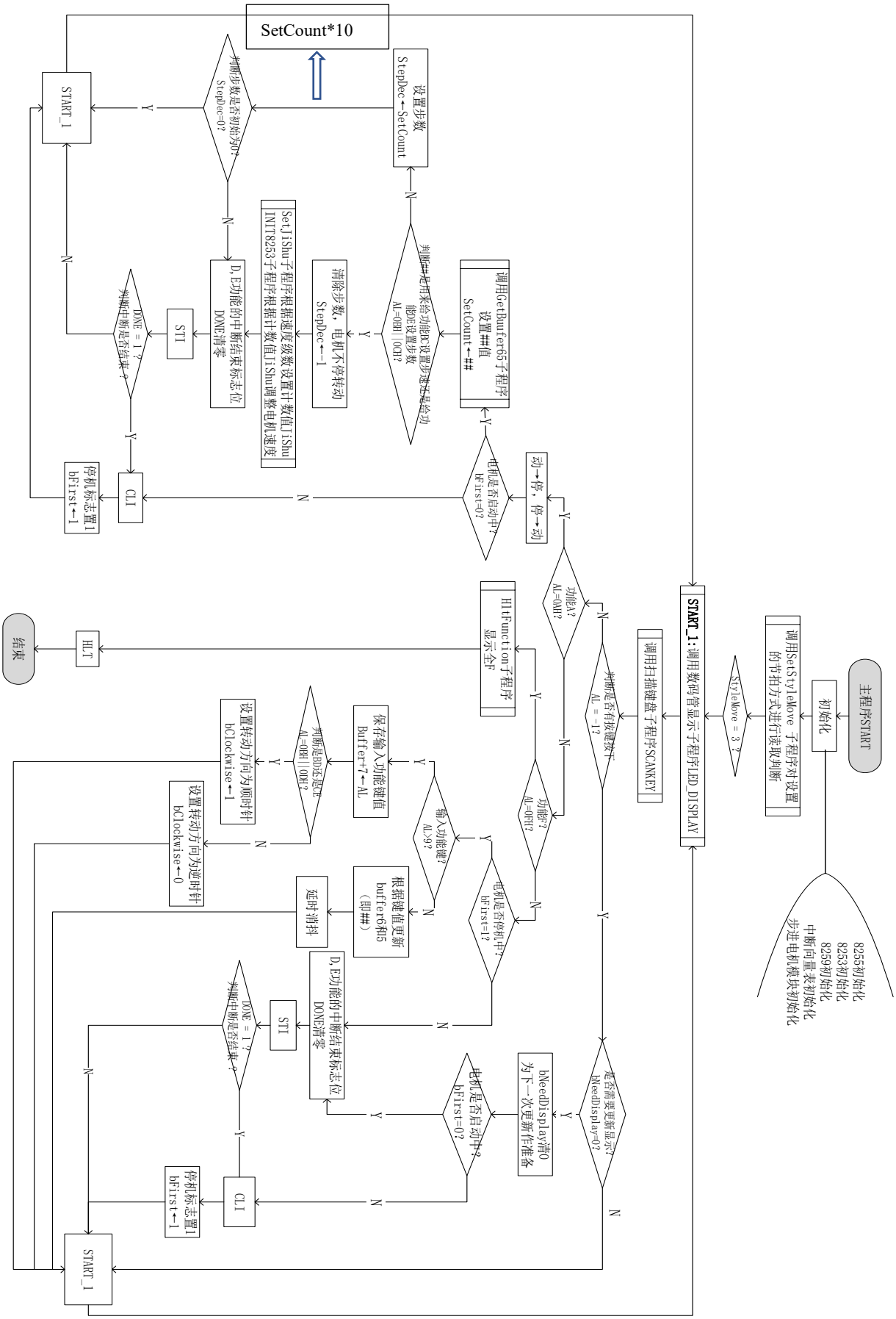

主程序START
初始化
8255初始化
8253初始化
8259初始化
中断向量表初始化
步进电机模块初始化
调用SetStyleMove 子程序对设置的节拍方式进行读取判断
StyleMove = 3 ?
START_1:调用数码管显示子程序LED_DISPLAY
调用扫描键盘子程序SCANKEY
判断是否有按键按下 AL = -1?
是否需要更新显示? bNeedDisplay=0?
bNeedDisplay清0 为下一次更新作准备
电机是否启动中? bFirst=0?
D,E功能的中断结束标志位 DONE清零
STI
DONE = 1 ?
判断中断是否结束 ?
CLI
停机标志置1 bFirst←1
START_1
电机是否停机中? bFirst=1?
延时消抖
根据键值更新 buffer6和5 (即##)
输入功能键? AL>9?
保存输入功能键值 Buffer+7← AL
判断是BD还是CE AL=0BH | 0CH?
设置转动方向为顺时针 bClockwise←1
设置转动方向为逆时针 bClockwise←0
功能? AL=0FH?
H1tFunction子程序 显示全F
HLT
结束
功能A? AL=0AH?
动→停，停→动
电机是否启动中? bFirst=0?
调用GetBuffer65子程序 设置##值 SetCount←##
判断##是用来给功能B设置步数还是给功能E设置步数 AL=0BH | 0CH?
清除步数，电机不停转动 StepDec←-1
SetJiShu子程序根据速度级数设置计数值JiShu INIT8253子程序根据计数值JiShu调整电机速度
设置步数 StepDec←SetCount
判断步数是否初始为0? StepDec=0?
SetCount*10
Y
N

图 12 主程序流程图（横）

5.2 初始化模块

5.2.1 INIT8253

选择计数器 0，工作在方式 2，采用二进制计数，通过改变计数值来改变步进电机的转速。

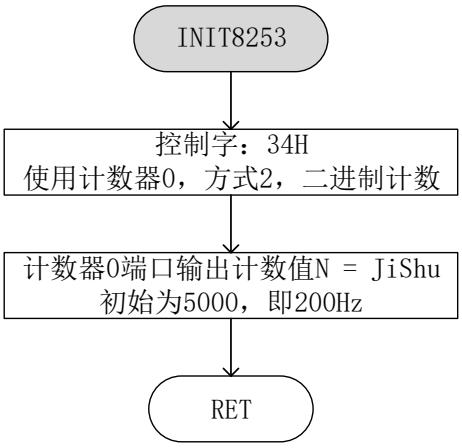


图 13 INIT8253 子程序

5.2.2 INIT8255

工作于方式 0，A 口输出，B 口输出，C 口低 4 位输入，高 4 位输出。

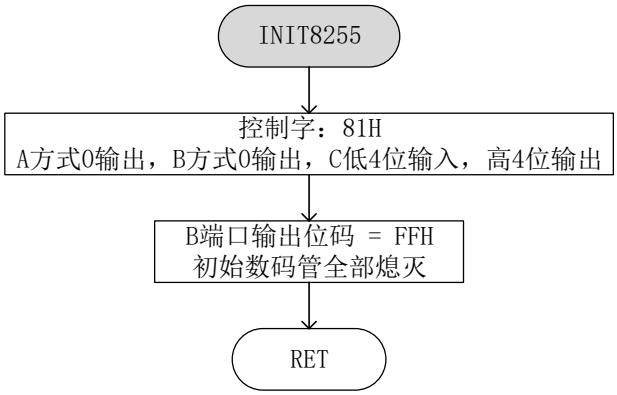


图 14 INIT8255 子程序

5.2.3 INIT8259

设置中断方式。

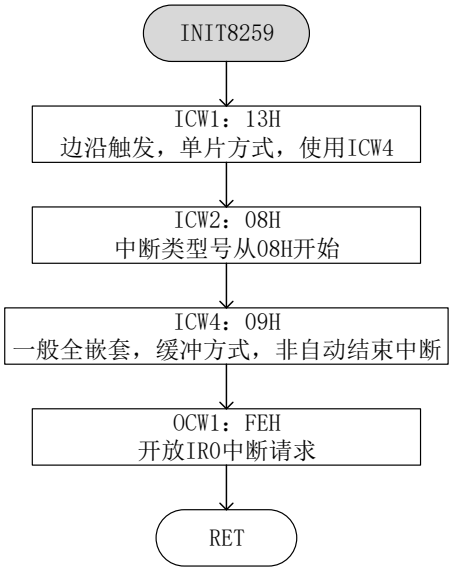


图 15 INIT8259 子程序

5.2.4 INITMOTOR

设置步进电机初始状态。

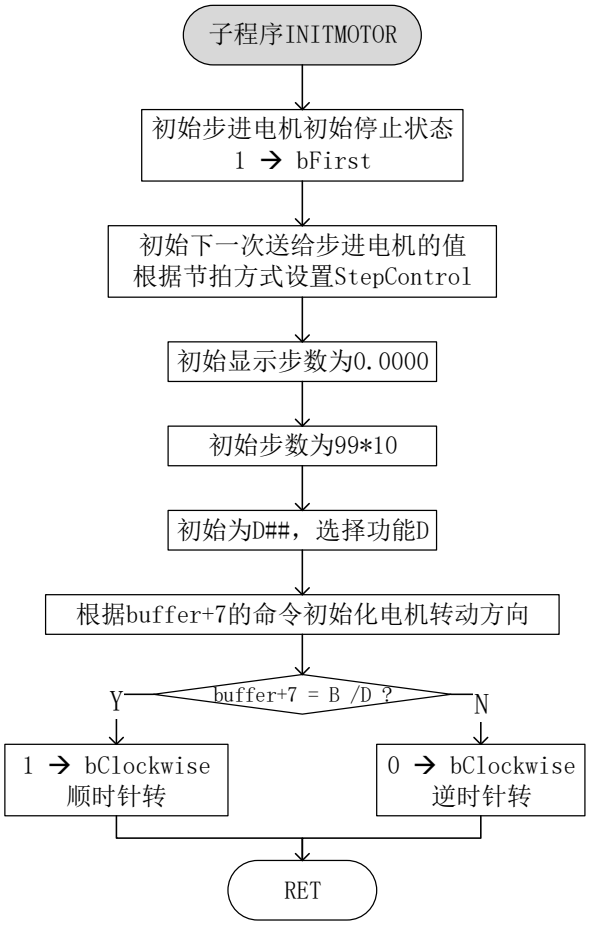


图 16 INITMOTOR 子程序

5.2.5 INTERRUPT_VECTOR

中断向量表初始化。

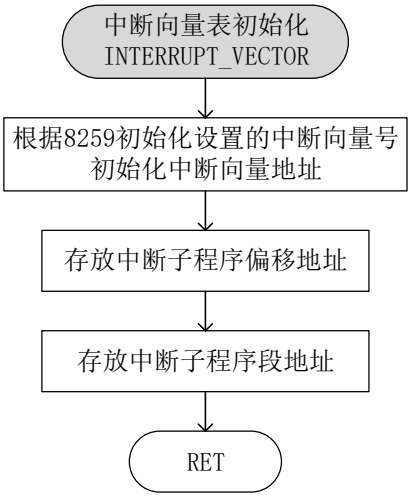


图 17 INIERRUPT_VECTOR

5.2.6 INIT8251

接收机和发送机初始化。

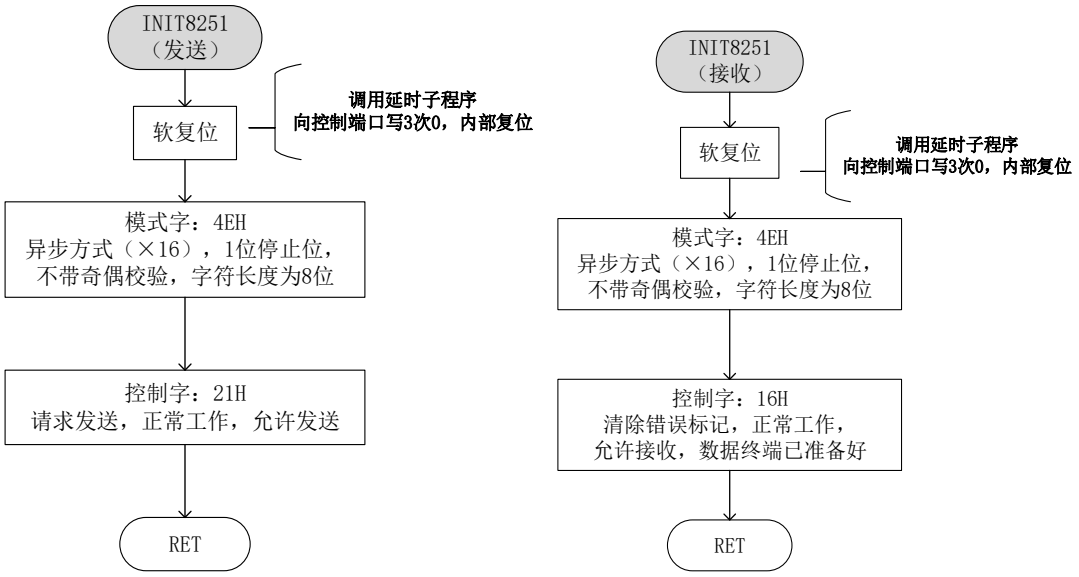


图 18 发送机 INIT8251

5.3 *SetStyleMove

设置电机转动方式。由于 8255 的 C 口低 4 位设置为输入，其中 PC1 和 PC0 用于控制键盘列线，PC3、PC2 悬空未用。因此可用 PC3、PC2 分别连接开关量 K6、K5，拨动开关将开关量通过 PC3 和 PC2 输入，存入 StyleMove 作为转动方式的选择。

表 6 设置步进电机的转动方式

K6	K5	意义
0	0	单 4 拍转动方式
0	1	双 4 拍转动方式
1	0	单双 8 拍转动方式
1	1	用来判断是否设置了转动方式

在主程序中通过比较 StyleMove 是否为 11，来判断是否设置了步进电机的转动方式：如果为 11，则循环等待设置转动方式，否则进行步进电机的转动。

转动方式	转动步骤（由 8255 中 PC 高四位控制）	
	正转	反转
单 4 拍	A→B→C→D→A 11H→22H→44H→88H→11H	C→B→A→D→C 44H→22H→11H→88H→44H
双 4 拍	AB→BC→CD→DA→AB 33H→66H→CCH→99H→33H	DC→CB→BA→AD→DC CCH→66H→33H→90H→CCH
单双 8 拍	A→AB→B→BC→C→CD→D→DA→A 10H→30H→20H→60H→40H→C0H→ 80H→90H→10H	B→BA→A→AD→D→DC→C→CB→B 20H→30H→10H→90H→80H→C0H→ 40H→60H→20H

图 19 步进电机转动方式

5.4 LED_DISPLAY

将键值查表转换成段码用于数码管显示。

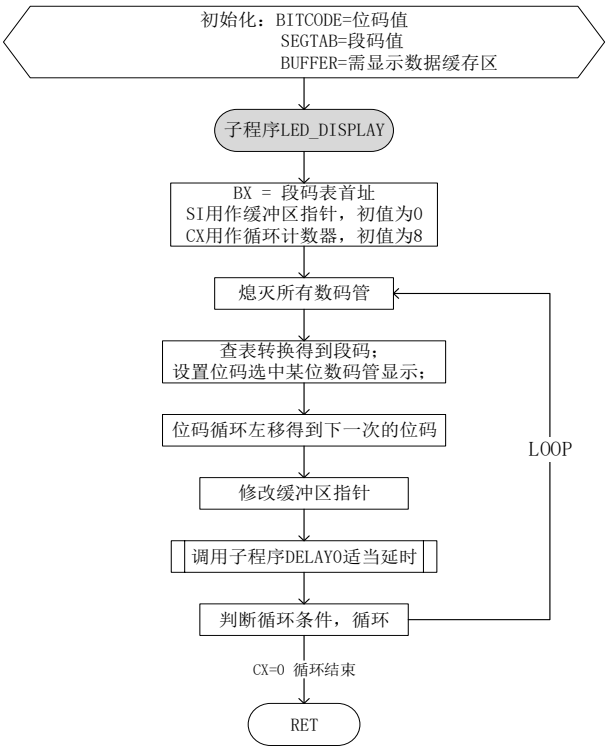


图 20 LED_DISPLAY 子程序

5.5 TRANSLATE

将行列码转换成键值。

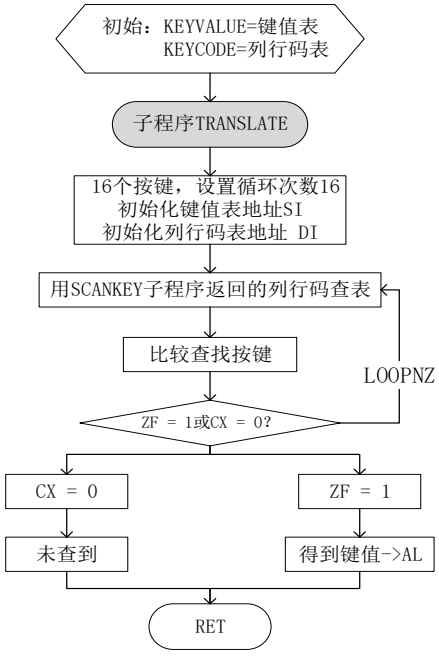


图 21 TRANSLATE 子程序

5.6 SCANKEY

扫描按键形成列行码，并调用 TRANSLATE 子程序将列行码转换成键值存放在 AL 中。

我修改的部分：由于循环调用 SCANKEY 子程序，为了保证数码管的正常显示，在 SCANKEY 中增加调用 LED_DISPLAY 子程序。

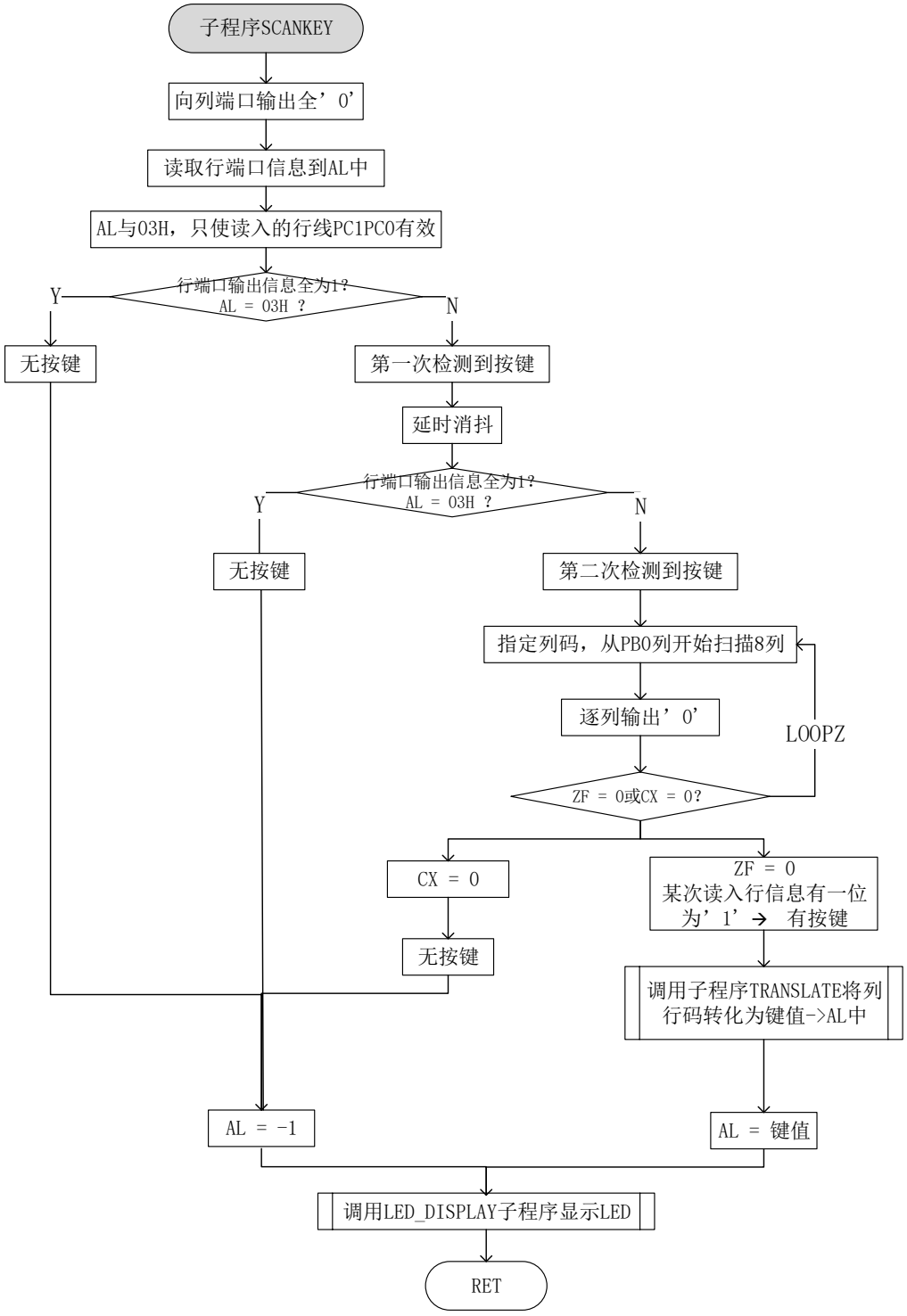


图 22 SCANKEY **子程序**

5.7 *GetBuffer65

由于输入到显示缓冲区中的数据是独立开的，即显示的两位数实际上并不是数学意义上的两位数，需要将 buffer6 和 buffer5 中的数据转换成数学意义上的两位数送入 SetCount 内存变量中：将 buffer+6 中的数据乘 10 作为十位数，再加上 buffer+5 中的数据。

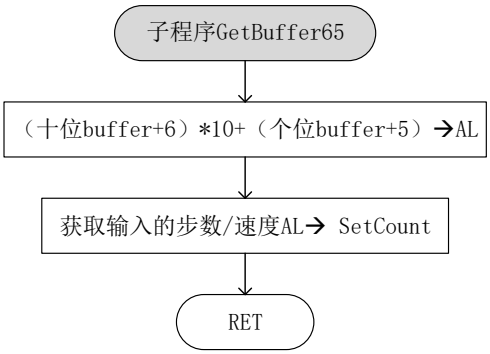


图 23 GetBuffer65 子程序

5.8 *SetJiShu

由于 SetCount 只用于存放功能参数，而应该是设置步数还是转速需要通过输入的键值来判断，并且步数和转速需要通过特定的公式计算。

改变步进电机转速通过调节脉冲的频率来实现。SetJiShu 子程序将输入的 SetCount 原始值加 1，送入速度级数内存变量 SpeedNumber 中。由于实际测试，电机转动频率有限制，直接设置 100 级速度可能会因为频率过快或过慢导致电机无法正常转动，因此需要对速度级数进行处理，使其设置每一级都能使电机转动。因此我们讨论通过公式进行 8253 计数值的处理，并且通过不断的调试，最终我们确定根据公式$5000 + 100 \times (101 - SpeedNumber)$计算出 8253 计数值送给内存变量 JiShu 中，每次设置新的转速都将改变产生中断的频率，从而实现步进电机转速的控制。

初始为200Hz，即0.005 秒/步
然后根据速度级数对应JiShu的范围15000~5100
则【单4拍和双四拍】对应的中断周期范围为0.015秒/步~0.0051秒/步
则【单双8拍】对应的中断周期范围为0.0075秒/步~0.00255秒/步

图 24 转动周期（中断周期）

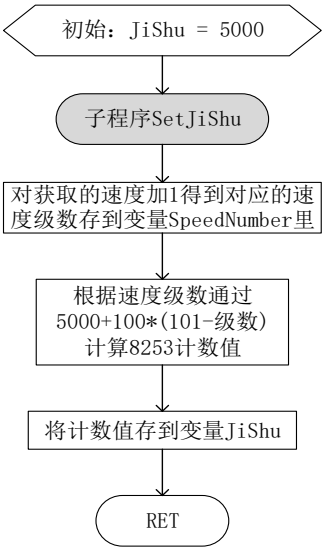


图 25 SetJiShu 子程序

5.9 *StepCountSet

与 GetBuffer65 实现的功能类似，StepCountSet 将当前步数的绝对值值转换成十进制形式保存在 StepCount 内存变量中，用于之后 AlterStep 子程序的调用。

$$StepCount = (buffer + 3) \times 1000 + (buffer + 2) \times 100 + (buffer + 1) \times 10 + \ (buffer)$$

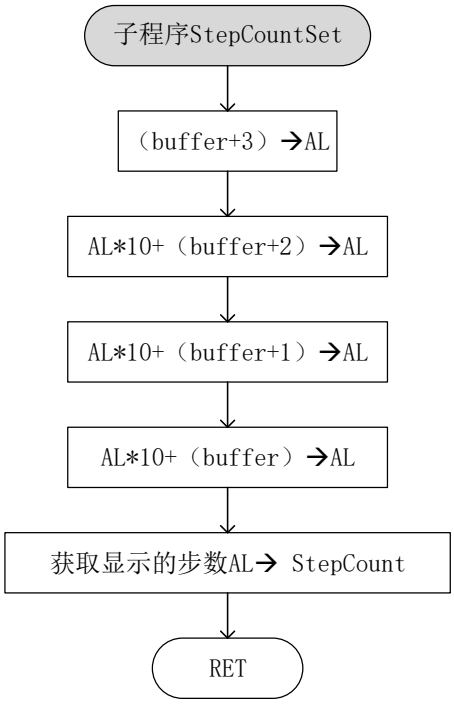


图 26 StepCountSet

5.10 *DELAY 系列

（1）DELAY0 子程序用于实现一定的延时，使多位数码管同时显示。

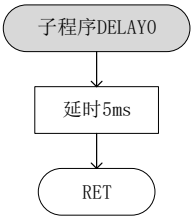


图 27 DELAY0 子程序

（2）DELAY1 子程序用于防止设置功能参数时输入数字过快。

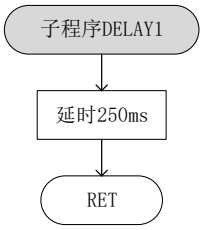


图 28 DELAY1 子程序

（3）DELAY2 子程序用于扫描按键时延时消抖。

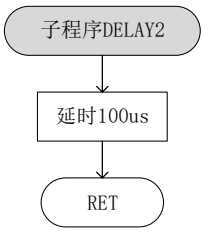


图 29 DELAY2 子程序

5.11 HltFunction

8 个数码管显示全“F”。

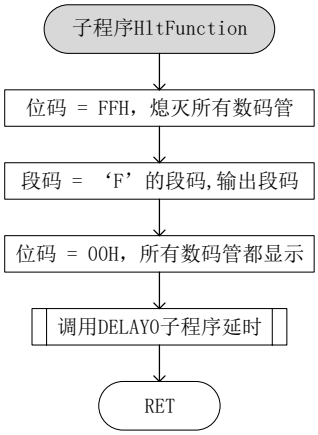


图 30 HltFunction 子程序

5.12 *StepControlSet

步进电机可按单四拍、双四拍、单双八拍等方式转动。StepControlSet 子程序用通过 SetStyleMove 子程序得到的转动方式偏移量 StyleMove 来判断电机的转动方式。

设置相位值数组 ControlStyle，由于单四拍和双四拍可通过左右移实现下一拍相位值的改变，因此只需要分别设置一个内存变量存放初始相位值即可。而单双八拍需要将各相相位值预先存入，通过改变数组指针来得到下一拍相位值。StyleMove 设置 0、1、2，分别指向单四拍、双四拍和单双八拍的初始相位值。

首先根据 StyleMove 区分是否是单双八拍需要特殊处理：

如果不是单双 8 拍，将当前相位值右移实现电机正转，左移实现电机反转；

如果是单双 8 拍，将指向 ControlStyle 的指针加 1 实现左移，减 1 实现右移。同时为了保证正常相位值产生，必须对指针进行“越界”判断：

若指针加 1 超出 10（即已实现电机反转，相位值一轮的转换），则将指针重新指向单双 8 拍转动方式下的数组第一位；

若指针减 1 小于 2（即已实现电机正转，相位值一轮的转换），则将指针指向单双 8 拍转动方式下的数组最后一位；

从而实现单双 8 拍方式下数组的循环。并将产生的下一步要送给步进电机的相位值保存在 StepControl 中。

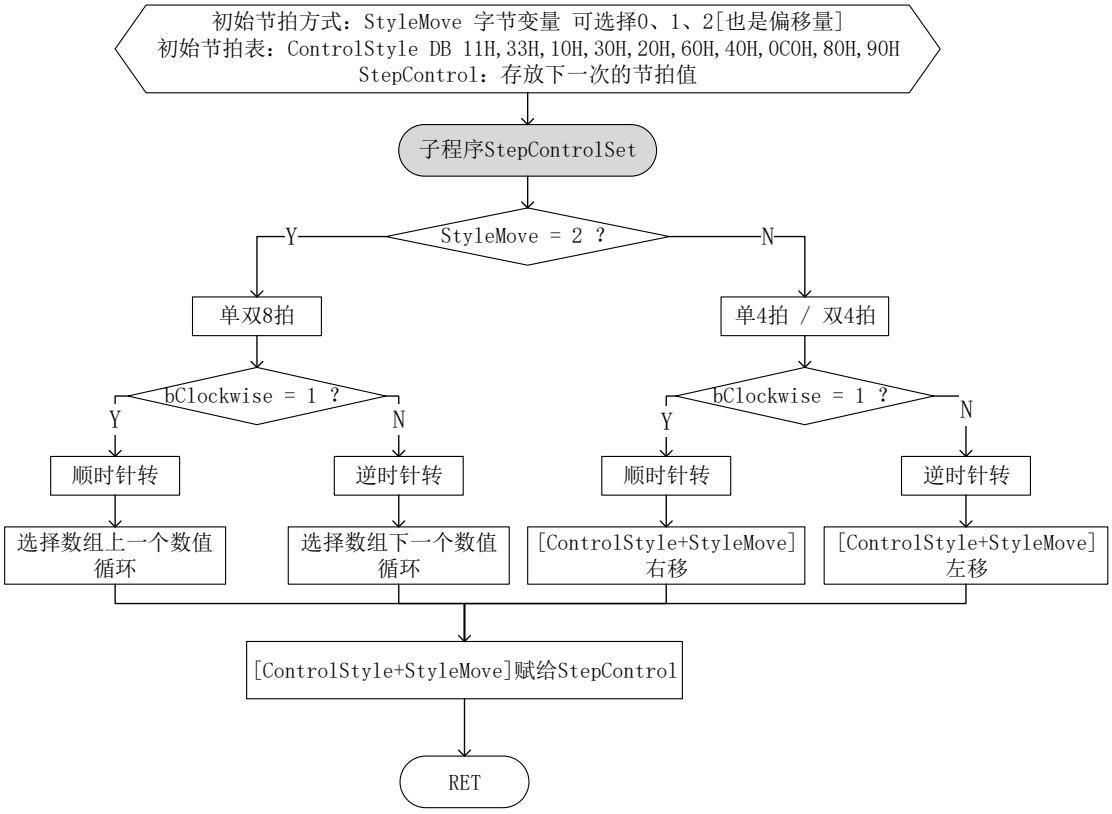


图 31 StepControlSet 子程序

5.13 *AlterStep

这是为了正常实现步数的显示设计的子程序，即正转一步将当前步数值加 1，反转一步将当前步数值减 1，应实现以下功能：

（1）判断此时电机的转动方向。

（2）电机正转方式下，判断此时步数显示是“正 0.xxxx”还是“负-.xxxx”；

1° 如果当前步数大于 0，那么每正转一步，显示的步数值应加 1；

2° 如果当前步数小于 0，那么每正转一步，显示的步数值应减 1；

3° 若当前步数为-1，即显示“-.0001”时，应将 buffer+5 中的“-.”改成“0.”表示步数由负转正，且将显示的步数值置 0，即下一步的显示应为“0.0000”。

（3）同理，在电机反转方式下，判断此时步数是正是负：

1° 若当前步数大于 0，则每反转一步，显示的步数值应减 1；

2° 若当前步数小于 0，则每反转一步，显示的步数值应加 1；

3° 若当前步数为 0，即显示“0.0000”时，应将 buffer+5 中的“0.”改成“-.”表示步数由正转负，且将显示的步数值置 1，即下一步的显示应为“-.0001”。

（4）当步数的绝对值为 9999，也就是显示“x.9999”时，表示已到正转（反转）下步数计数的最大值，我们设置此时步进电机进行反转，即：

1° 显示“0.9999”时，步机由正转改为反转，并将显示功能键值的 buffer+7 进行相应的修改——“b”转为显示“C”，“d”转为显示“E”；

2° 同理，显示“-.9999”时，步机由反转改为正转，并 buffer+7 中的功能键值进行对应的修改——“C”转为显示“b”，“E”转为显示“d”。

判断步数的绝对值是否已到达 9999，由调用 StepCountSet 子程序得到的 StepCount 内存变量与 9999 进行比较。

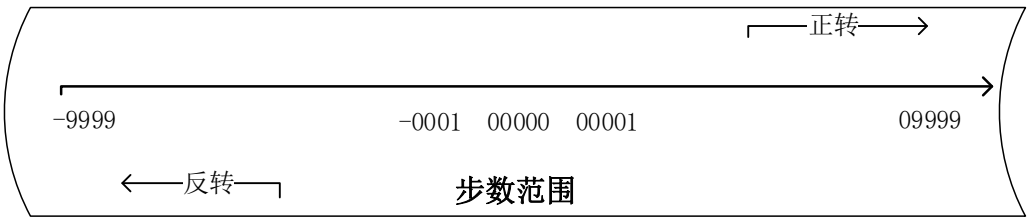


图 32 步数表示范围

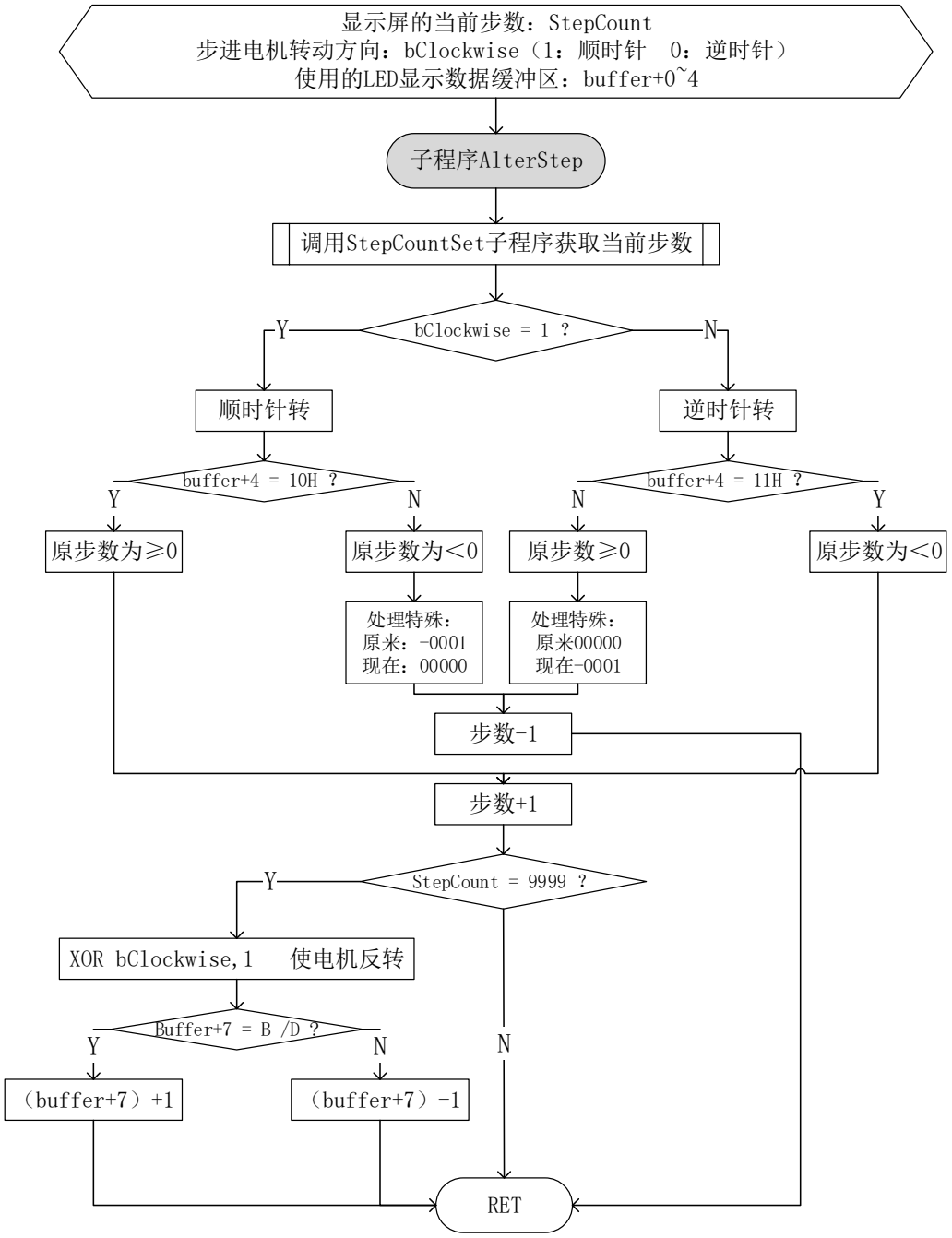


图 33 AlterStep 子程序

5.14 *TIMERO

首先通过 8255 的 C 口高 4 位将相位值送给步进电机，控制步进电机进行一步的转动。其次调用 StepControlSet 子程序产生下一步应送给步进电机的相位值 StepControl。接着调用 AlterStep 子程序更新当前步数的数码管显示，并置数码管更新标志 bNeedDisplay 为 1，即代表已转动一步，需要更新步数。

通过比较 StepDec 是否为-1 来判断此时步进电机的功能到底是设置了步数还是设置转速：

（1）若通过功能键 D、E 设置了步进电机的转速，除非键入 A，否则步进电机以设置的转速进行匀速转动；

（2）否则，通过功能键 B、C 设置了步进电机的步数。那么当走完指定的步数之后，步进电机应停止运行。通过比较 StepDec 是否为 0 来判断当前设置的步数是否已走完：

1°若 StepDec 还未减至 0，则代表步数未走完，结束这一步的中断，继续下一步的转动；

2°若 StepDec 为 0，则代表步数已走完，此时应关中断 CLI，禁止下一步执行，并将电机置停机状态，置停机标志 bFirst 为 1。

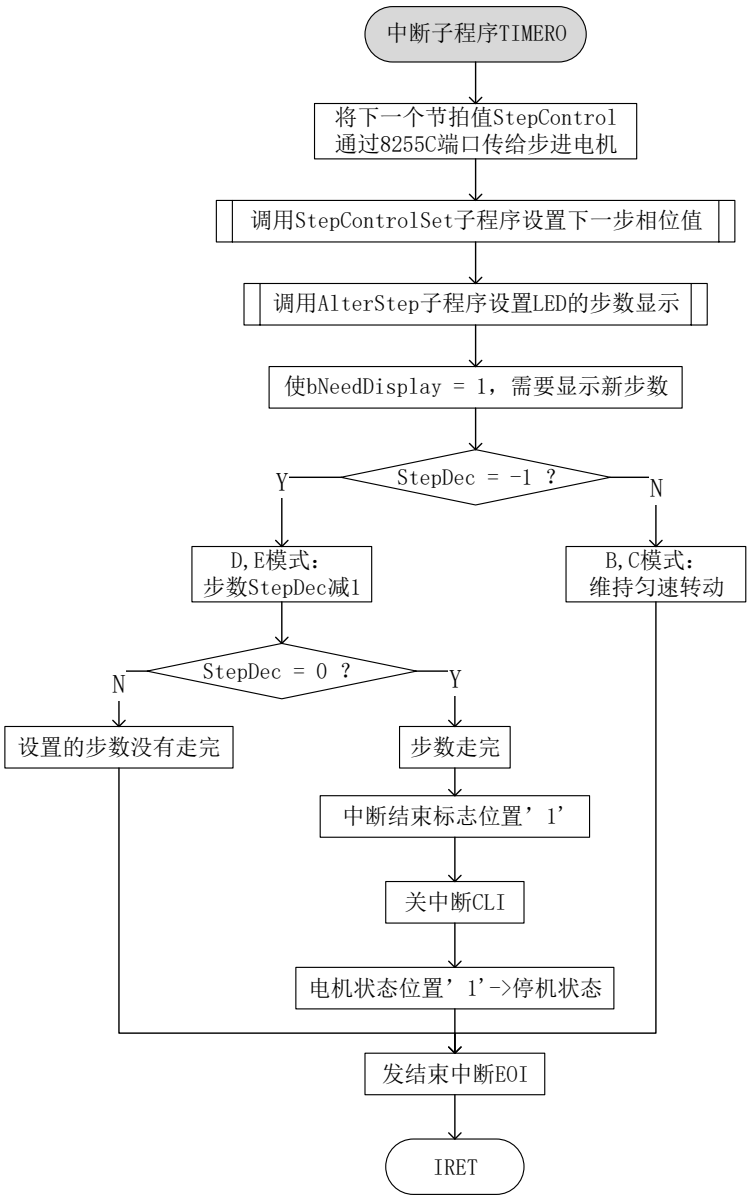


图 34 TIMER0 中断子程序

5.15 SEND8251

发送机查询 TxRDY 是否有效，通过 8251 将 buffer 中的数据发送给接收机。

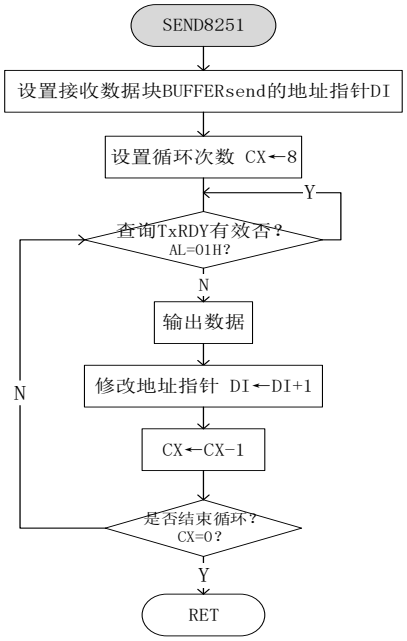


图 35 SEND8251 子程序

5.16 RECEIVE8251

接收机查询 RxRDY 是否有效，通过 8251 将发送机发送的数据接收并保存在自身的 buffer 数组中。

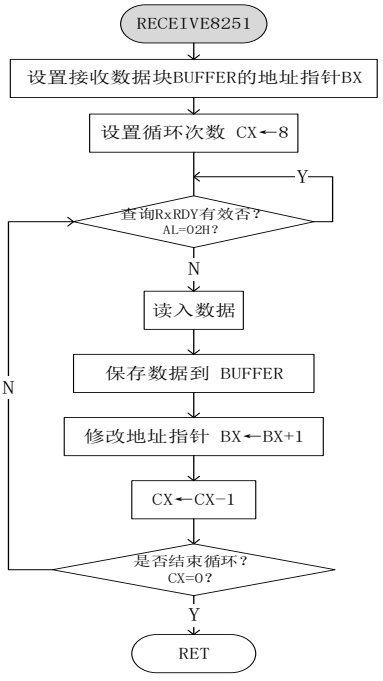


图 36 RECEIVE8251 子程序

六、 软硬件调试过程

6.1 调试过程

（1）首先测试所用连线是否通路。

（2）关闭电源后，硬件连线。

（3）成功编译.asm 文件后，装载.dob 文件，进行单步测试。

根据单步测试的每一步结果，观察是否符合预期需求：

若符合预期所想，进行第（4）步；

否则，针对错误的结果，对源程序进行相应的修改，重新编译加载程序，进行第（4）步，直到符合预期为止。

（4）全速运行程序，观察是否符合预期需求；

若在出现问题的地方停止运行，观察寄存器和变量窗，分析问题，针对错误的结果，对源程序进行相应的修改，重新编译加载程序，进行第（4）步，直到符合预期为止。

若符合预期所想，进行第（5）步；

（5）思考是否可以对其进行改进，若对其改进和创新，跳转到第（4）步，否则程序测试成功并已完成。

6.2 问题与解决方法

1. PC 指针超出范围

由于一些子程序忘记写 RET，虽然可以编译程序，但会出现指针跑飞的现象，出现 PC 指针超出范围的错误。

【解决方法】仔细检查源程序，加上 RET。

2. 设置功能参数问题

功能参数为两位数，当只键入一次数字值时，会出现两个数码管上同时修改为键入值，比如设置“10”，会出现设置成“11”的情况。

【解决方法】在键入 0~9 时进行适当的延时。

3. 步数误差问题

当电机处于 DE 功能下时，理应在指定步数走完后停止，实际发现电机总是比设置的步数多走了几步之后才会停止。

【原因分析】经过多次调试，我们猜测，应该与中断频率有关。当转速较慢时，误差较小；转速较快时，误差较大。中断速度太快，导致主程序判断不及时，必然存在步数误差。

4. 转速问题

当设置的转动频率过快或过慢时，步进电机将处于停止状态。

虽然速度可设置 1~100 级，但变化幅度并不大。

【解决方法】通过合理的数学公式（已在前文给出）将设置的速度级数转换成合理的计数值，使电机可在 1~100 级的速度下转动。

5. 跳转范围越界

在调试过程中，出现过主程序太长，导致无法通过条件转移指令跳转到相应位置的问题，即超出了条件转移指令的下一条指令到目的地址之间的相对位移量（必须在-128~+127 内）。

【解决方法】在不影响主程序逻辑结构的前提下，将缩短条件转移指令和跳转模块的距离。

6. 按键控制子程序问题

一开始我们的设计思想是：扫描按键后，通过调用 ChooseControl 子程序进行功能的判断和设置。但在实际调试过程中，总存在问题导致电机无法按照指定的功能正常启停。

【解决方法】将 ChooseControl 子程序的功能直接放入主程序中。

7. 停机问题

（1）步进电机无法通过按键 A 正常停止转动，之后也无法通过按键 A 正常启动。

（2）DE 功能下，电机在走完指定步数后仍不予停止。

【解决方法】设置停机标志 bFirst 判断电机的运行状态，将对 StepDec 的判断移入中断子程序 TIMERO 中。

8. 转动方向问题

以单 4 拍为例，按照原理理解，正转应为 A→B→C→D→A，但在实际运行过程中，正转却为 C→B→A→D→C。

【解决方法】以实际硬件操作为主。

9. 显示问题

起初，B 端口既是列线又是位码输出，在进行列扫描时会向列端口输出全“0”测试有无按键按下，此时导致 8 个数码管都被选中显示全 0。

【解决方法】在 SCANKEY 扫描按键子程序中调用 LED_DISPLAY 数码管显示子程序。

此外，为了使多个数码管同时显示，必须进行一定时间的延时，否则将会出现数码管从左到右依次显示的问题。

10. 键入 A 问题

在调试时我们发现，有时按 A 电机没有反应，需要再按一两次电机才会启停。

【原因分析】可能存在按键抖动，可适当加入一定的延时，但仍存在按键抖动现象。而且如果加入的延时不恰当，会出现数码管屏闪且显示全“8”的问题。

11. 当前步数不同步问题

无论转速如何变化，当前步数的改变频率总是不变，与实际步数的改变不同步。

【解决方法】在中断子程序 TIMERO 内进行当前步数的更新。

七、 设计总结

7.1 已实现的功能

（1）能够扫描并显示按键，可以根据相应功能键值和设置的参数对步进电机进行控制，同时可通过数码管显示当前步数，满足基本选题要求。

（2）能够使步进电机在跨度较大的速度范围（级数：1~100）下运行；

（3）BC 模式下，步进电机当前步数绝对值计数达到最大（9999）时，实现电机反转，并将反转后的功能键值显示在数码管上。

（4）在步进电机启动状态下，实现屏蔽键值的效果：只有键入 A 能控制电机启停，键入 F 结束整个程序；其他按键在电机转动的情况下，即使键入也视作无效，对步进电机的状态设定无影响。

（5）实现通过开关量设置步进电机的转动方式，步进电机在单 4 拍、双 4 拍和单双 8 拍三种转动方式下都可运行。

（6）通过 8251 实现系统之间的通信：发送机发送数据初始化接收机的电机状态。

7.2 待改进的地方

7.2.1 可以使步进电机实现变速转动

目前系统只能够在 BC 模式下按照设定的速度匀速运行，而不能够实现变速功能。

7.2.2 步数问题

（1）DE 模式下的步数误差：

目前系统在 DE 模式下设置一定的步数让电机停下来，但实际显示的步数要大一些，有一定的误差。

（2）步数跳动过快问题：

为每一步计数使得当前步数的最后一位跳动很快，肉眼几乎不能辨认个位步数的显示。

7.2.3 按键抖动问题

目前系统在按键“A”的操作上有仍有按键抖动的问题，有时不能正确地对电机启停。

7.3 改进的措施

- 可实现变速转动：

一种是若设置在电机启动状态下按除“A”“F”外的按键有效，则可在 B,C 模式下修改速度，使得步进电机变速运行，但这种方式对步进电机的影响较大，若设置的速度相差较大则电机有可能会卡住不转。

另一种可以通过 0809 对 0~5V 电压进行模数转换，然后将其持续循环的输入给 8253 模块，修改其频率达到变速的效果。这种方式下速度不会突然跳变，对电机的影响较小。

- 每十步计 1 次数解决步数跳动太快的问题：

可以每十步计 1 次数，不仅当前步数值显示会更清晰，而且扩大了步数的计数范围。

7.4 创新方法

7.4.1 电机反转

我们设置若当前步数的绝对值已到达 9999 步计数上限时，改变电机的转动方向，同时修改显示的功能键值。

7.4.2 开关量设置转动方式

通过拨动 K5、K6 可设置电机的转动方式，是单四拍、双四拍还是单双八拍。

7.4.3 通过 8251 通信

利用 8251 可以实现步进电机之间的控制：发送机将自身 buffer 缓冲区中的数据送给接收机，接收机成功接收后保存在自身的 buffer 缓冲区中，此后根据 buffer 中的数据设定功能和功能参数，按 A 进行转动。

7.4.4 0809 控制、蜂鸣器提示

在实现了 8251 通信的前提下，将发送机设置为主机，将接收机设置为从机。主机设定不变，从机修改为：

（1）无法通过按键设定功能和参数，只能通过主机发送指定的功能和参数控制从机步进电机进行转动；

（2）通过 ADC0809 对旋转黑色旋钮产生的电压进行模数转换，当电压为 5V 时，控制从机电机停止，从而实现用旋钮控制从机电机启停的功能：

1°只有在从机电机停止的情况下，才能通过 8251 通信，接收主机发来的数据；

2°只有在主机启动状态下，按下任意键，将数据传送给从机。

（3）此时因为不需要按键控制，8255 的 C 口低 4 位全部未用，此时可将 C 口低 4 位设置成方式 0 输出。将 PC0 连接到蜂鸣器的 Ctrl 端口，实现功能如下：

1°当从机电机工作在 DE 模式下，走完设定步数时，通过 PC0 发信号给 Ctrl，提示步数已走完；

2°旋钮控制从机停止后，通过蜂鸣器进行提示。

八、 课程设计的收获及心得体会

8.1 课程设计的收获

1、对 8255、8253、8259、8251、0809 等芯片有了更深层级的认知和理解；

2、对步进电机的步进式旋转有了一定的了解；

3、通过思考和探究，极大地锻炼了逻辑思维，提高了编程能力；

4、培养了团队意识和责任感；

5、在不断的调试过程中，领悟到失败不是最后的结果，只是通往成功塔的荆棘阶梯。我们要做的不是停滞不前或者倒退，而且尝试作出新的改变、拓展新的思路，寻找踏上阶梯的那一步。

6、星研调试程序时，进入中断后单步运行无法跳出。后来发现的一个解决方法是：开中断还未进入中断之前，全速断点程序，可查看此时变量窗中变量的变化情况。

8.2 心得体会

为期一周的课设终于即将迎来“成功通关”的提示，一开始我们想着，既然是一种能力的体现，一个锻炼的舞台，不如就试着挑战吧！并且由于电子钟我们之前数字逻辑的课设已经完成过，于是我们放弃了电子钟，转而投入对步进电机的研究中。

一开始就遇到了不小的挫折。在前三天的设计和调试过程中，虽然不能说没有进展，但是进度非常慢。找不到问题，也想不出解决方案，连晚上做梦都想着步进电机究竟怎么转。曾想过：“要不就这样算了吧，何必折腾自己呢？”最后还是顺了顺头发:“做都做了，弄不出来就放弃，这不行啊。”其实只要坚持下去，阳光总在风雨后。我们简短地讨论之后，决定加班加点，偷偷跟着其他班一起做，功夫不负有心人，终于在第四天有了重大的突破，随后开始头脑风暴，思考应该从何改进，从何处入口。

起先，我们考虑用蜂鸣器实现 DE 模式下步数走完后发出提示的功能，同时

当实在是卡在了瓶颈，进退维谷之时，不要害羞不要害怕，可以询问老师意见，虽然不是一个选题也可以参考同学的解决方案，寻求灵感。自己苦思冥想反而是一种浪费时间的行为，当然思考的时间一点都不可以省。

这次的课程设计留下了遗憾，最后灵感迸发想要实现的功能只能是一次性的，第二次之后 0809 和 8251 的配合就出现了问题，很遗憾。

感谢我的队友，富有责任感，配合默契，心有灵犀，能在关键时刻点醒我，我们一起认真思考和探讨，虚心采纳对方的意见。团队合作，事半功倍！

九、 附录：源程序清单

9.1 主程序

```
.MODEL TINY
PORT8255_A    EQU 270H              ;CS1
PORT8255_B    EQU 271H
PORT8255_C    EQU 272H
PORT8255_K    EQU 273H
PORT8253_0    EQU 260H              ;CS2
PORT8253_1    EQU 261H
PORT8253_2    EQU 262H
PORT8253_K    EQU 263H
PORT8259_0    EQU 250H              ;CS3
PORT8259_1    EQU 251H
.STACK 100
.DATA
    BITCODE          DB        ?                              ;存放位码值
    SEGTAB                DB          0C0H,0F9H,0A4H,0B0H,99H,92H,82H,0F8H,80H,90H,
88H,83H,0C6H,0A1H,86H,8EH,40H,3FH              ;段码->LED 灯编码（0~F+"0."+"-."）
    KEYVALUE                                                                DB
00H,01H,02H,03H,04H,05H,06H,07H,08H,09H,0AH,0BH,0CH,0DH,0EH,0FH
  ;键值表 +【10H -> "0."】+【11H -> "-."】
    KEYCODE                                                                 DW
0FE02H,0FD02H,0FB02H,0F702H,0EF02H,0DF02H,0BF02H,7F02H,0FE01H,0FD01H,0FB01
H,0F701H,0EF01H,0DF01H,0BF01H,7F01H    ;键盘列行码
    BUFFER           DB        8 DUP(?)                        ;数码管上显示内容缓存区
    bFirst           DB        0                       ;启停步进电机（0：启动；1：停止）
    bClockwise       DB        0                       ;电机转动方向（1：顺时针；0：逆时针）
    bNeedDisplay     DB        0                       ;已转动一步，需要显示新的步数
    StepControl      DB        0                             ;给步进电机下一步的相位值
    SetCount         DB        0                              ;步进电机的步数/转速
    StepCount        DW        0                              ;数码管显示的步数值
    StepDec          DW        0                              ;D,E 设置的步数
    SpeedNumber      DB        0                              ;B,C 设置的转速
    JiShu            DW        5000                    ;8253 计数初值 5000，0.005 秒/步
    StyleMove        DB        0                       ;方式
    ControlStyle     DB        11H,33H,10H,30H,20H,60H,40H,0C0H,80H,90H          ;拍值
    DONE             DB        0                       ;设置的步数已走完及中断完成标志位
.CODE
START:
    MOV AX,@DATA
```

```
    MOV DS,AX                    ;装载 DS
    ;-------------初始化部分
    CALL INIT8251                ;8251 初始化
    CALL INIT8253                ;8253 初始化
    CALL INIT8255                ;8255 初始化
    CALL INIT8259                ;8259 初始化
    CALL INTERRUPT_VECTOR        ;中断向量表初始化
    ;（1）CALL SEND8251 接收机使用
    ;（2）CALL RECEIVE8251 发送机使用
    CALL INITMOTOR               ;步进电机模块初始化
    ;-------------等待设置步进电机节拍方式
WaitingSet:
    CALL SetStyleMove            ;调用子程序对设置的节拍方式进行读取判断
    CMP StyleMove,3              ;判断是否设置了节拍方式
    JZ WaitingSet                ;还未设置节拍方式继续等待
    ;-------------开始显示并扫描按键
START_1:
    CALL LED_DISPLAY             ;调用数码管显示子程序
WAITEKEY:
    CALL SCANKEY                 ;调用扫描键盘子程序
    CMP AL,-1                    ;判断是否有按键按下
    JNZ START_2                  ;有按键按下，则跳转开始判断按键
    CMP bNeedDisplay,0           ;无按键，则判断电机有没有转动
    JZ   WAITEKEY                ;无按键，电机没有转动，则继续等待按键
    MOV bNeedDisplay,0           ;清零为下一次转动做准备
    CMP bFirst,0                 ;电机转动了一步，且此时没有按键按下，判断电机此时的状态
    JZ   Exec1                   ;电机处于启动状态-->则开中断启动
    JMP StopNEXT                 ;电机处于停机状态-->则关中断停止
    ;-------------判断按键+设置参数
START_2:                         ;开始判断按键
    CMP AL,0AH                   ;如果输入 A
    JNZ START_OTHER              ;若果不是 A 跳转到执行
    XOR bFirst,1                 ;启停步进电机
    CMP bFirst,0                 ;判断是启动还是停止步进电机
    JNZ StopNEXT                 ;bFirst=1，跳转到 StopNEXT
StartNEXT:                       ;bFirst=0，启动步进电机
    CALL Getbuffer65             ;设置 SetCount
    CMP buffer+7,0BH             ;如果是功能 B
    JZ BC                        ;跳转到 BC，设置步速
    CMP buffer+7,0CH             ;如果是功能 C
    JZ BC                        ;跳转到 BC，设置步速
DE:                              ;功能 DE，设置步数
    MOV AL,SetCount              ;AL <-- SetCount
    MOV BL,10                    ;BL <-- 10
```

```
    MUL BL                    ;设置步数为 10*SetCount
    MOV StepDec,AX            ;再将步数送给 StepDec(即输入 99，设置步数为 990 步)
    CMP StepDec,0             ;若初始设置为 0
    JZ   START_1              ;跳转到开始继续等待按键
    JMP Exec1                 ;否则开中断
BC:                           ;功能 BC，设置步速
    MOV StepDec,-1            ;清除步数
    CALL SetJiShu             ;调用子程序设置 8253 计数值
    CALL INIT8253             ;设置新的速度
Exec1:                        ;开始启动步进电机
    MOV DONE,0                ;D,E 功能下，中断结束标志位清零
    STI                       ;开中断
    CMP DONE,1                ;判断中断是否结束
    JNZ START_1               ;没有结束跳转到开始继续扫描按键等待
StopNEXT:                     ;bFirst=1，停止步进电机
    CLI                       ;关中断
    MOV bFirst,1              ;再赋给 bFirst 一次值
    JMP START_1               ;跳转到开始继续扫描按键
START_OTHER:                  ;如果输入 0~9 或 B~F
    CMP AL,0FH                ;如果输入 F
    JZ F                      ;跳转到 F 使步进电机停机
    CMP bFirst,1              ;判断电机当前状态
    JZ SHURU                  ;电机处于停机状态，则跳转设置 0~9，B~E
    JMP Exec1                 ;电机处于启动状态，则除 A,F 按键外其他按键不能进行操作
SHURU:
    CMP AL,9                  ;如果输入 0~9
    JA START_BCDE             ;将键值送入 buffer+6 和 buffer+5，即步数/步速
    MOV AH,buffer+5           ;根据输入的键值从右到左进行步速/步数更新
    MOV buffer+6,AH
    MOV buffer+5,AL
    CALL DELAY1               ;调用延时子程序：防止按键输入过快
    JMP START_1               ;否则跳转显示刚才设置的数据或命令,然后继续等待按键
START_BCDE:                   ;如果输入 B,C,D,E
    MOV buffer+7,AL           ;将 B,C,D,E 的功能键值送入 buffer+7
    CMP AL,0BH                ;如果输入 B
    JZ BD                     ;跳转到 BD，设置转动方向（顺时针）
    CMP AL,0DH                ;如果输入 D
    JZ BD                     ;跳转到 BD，设置转动方向（顺时针）
    MOV bClockwise,0          ;否则为功能 CE，设置转动方向为逆时针
    JMP START_1
BD:                           ;功能 BD，设置为顺时针
    MOV bClockwise,1
    JMP START_1               ;跳转到开始继续扫描按键等待
F:                            ;F--->程序结束
```

```
    CLI                          ;关中断
    CALL HltFunction             ;调用停机显示‘F’子程序
    HLT                          ;停机
```

9.2 初始化模块

9.2.1 INIT8253

```
;子程序名：INIT8253
;功能：8253 初始化
INIT8253 PROC NEAR
    MOV DX,PORT8253_K
    MOV AL,34H                   ;00110100 计数器 0，方式 2，二进制计数
    OUT DX,AL
    MOV DX,PORT8253_0
    MOV AX,JiShu                 ;10000+50*(101-n)
    OUT DX,AL
    MOV AL,AH
    OUT DX,AL
    RET
INIT8253 ENDP
```

9.2.2 INIT8255

```
;子程序名：[INIT8255]
;功能：8255 初始化
INIT8255 PROC NEAR
    MOV DX,PORT8255_K
    MOV AL,81H                   ;A 方式 0 输出，B 方式 0 输出，C 低 4 位输入，高 4 位输出
    OUT DX,AL
    MOV   DX, PORT8255_B
    MOV AL,0FFH                  ;位码全 1，初始数码管全部熄灭
    OUT DX,AL
    RET
INIT8255 ENDP
```

9.2.3 INIT8259

```
;子程序名：[INIT8259]
;功能：8259 初始化
INIT8259 PROC NEAR
    MOV DX,PORT8259_0
```

```
    MOV AL,13H                  ;[ICW1]00010011,边沿触发，单片方式，使用 ICW4
    OUT DX,AL
    MOV DX,PORT8259_1
    MOV AL,08H                  ;[ICW2]00001000,中断类型号从 08H 开始
    OUT DX,AL
    MOV AL,09H                 ;[ICW4]00001001,一般全嵌套，缓冲方式，非自动结束中断
    OUT DX,AL
    MOV AL,0FEH                 ;[OCW1]11111110,开放 IR0 中断请求
    OUT DX,AL
    RET
INIT8259 ENDP
```

9.2.4 INITMOTOR

```
;子程序名：INITMOTOR
;功能：步进电机模块初始化+BUFFER 初始化
INITMOTOR PROC NEAR
    MOV bFirst,1                        ;初始步进电机初始停止状态
    MOV BL,StyleMove                      ;将偏移量送入 StyleMove
    MOV AL,[ControlStyle+BX]
    MOV StepControl,AL                  ;初始下一次送给步进电机的值
    MOV buffer,0                        ;初始化显示 D990.0000
    MOV buffer+1,0
    MOV buffer+2,0
    MOV buffer+3,0                      ;初始显示步数为 0.0000
    MOV buffer+4,10H
    MOV buffer+5,9                      ;初始步数为 990
    MOV buffer+6,9
    MOV buffer+7,0DH                     ;初始为 D##
    CMP buffer+7,0BH                    ;根据 buffer+7 的命令初始化电机转动方向
    JZ BDinit
    CMP buffer+7,0DH
    JZ BDinit
    MOV bClockwise,0
BDinit:   MOV bClockwise,1
    RET
    INITMOTOR ENDP
```

9.2.5 INITERRUPT_VECTOR

```
;子程序名：[INTERRUPT_VECTOR]
;功能：中断向量表初始化
;影响寄存器：ES,AX,BX
```

```
INTERRUPT_VECTOR PROC NEAR
    PUSH ES
    PUSH AX
    PUSH BX
    MOV AX,0
    MOV ES,AX
    MOV BX,08H*4            ;BX<-中断向量地址
    MOV AX,OFFSET TIMERO    ;中断子程序地址
    MOV ES:[BX],AX          ;存放偏移地址
    MOV AX,SEG TIMERO
    MOV ES:[BX+2],AX        ;存放段地址
    POP BX
    POP AX
    POP ES
    RET
INTERRUPT_VECTOR ENDP
```

9.2.6 INIT8251

1、发送机

```
;子程序名：[INIT8251]
;功能：8251 初始化
INIT8251 PROC NEAR
    ;--------软复位----------
    MOV CX,3                ;
    XOR AL,AL               ;
    MOV DX,PORT8251_1       ;
AGA:OUT DX,AL               ;连续写入 3 个 00H
    CALL DELAY0             ;每次写入后延时一定的时间
    LOOP AGA                ;
    MOV AL,40H              ;
    OUT DX,AL               ;写入 40H
    CALL DELAY0
    ;------8251 初始化--------
    MOV AL,4EH              ;01001110B 设置方式字【异步×16，数据位 8 位，不带奇
偶校验位，1 位停止位】
    OUT DX,AL               ;
    CALL DELAY0
    MOV AL,21H              ;00100001B 设置控制字【允许发送，请求发送】
    OUT DX,AL               ;
    RET
INIT8251 ENDP
```

2、接收机

```
;子程序名：[INIT8251]
;功能：8251 初始化
INIT8251 PROC NEAR
    ;--------软复位----------
    MOV CX,3                    ;
    XOR AL,AL                   ;
    MOV DX,PORT8251_1           ;
AGA:OUT DX,AL                   ;连续写入 3 个 00H
    CALL DELAY0                 ;每次写入后延时一定的时间
    LOOP AGA                    ;
    MOV AL,40H                  ;
    OUT DX,AL                   ;写入 40H
    CALL DELAY0
    ;------8251 初始化--------
    MOV AL,4EH                  ;01001110B 设置方式字【异步×16，数据位 8 位，不带奇
偶校验位，1 位停止位】
    OUT DX,AL                   ;
    CALL DELAY0
    MOV AL,16H                  ;00010110H 设置控制字【清除错误标志，允许接收，数据
终端准备好】
    OUT DX,AL                   ;
    RET
INIT8251 ENDP
```

9.3 SetStyleMove

```
;子程序名：【SetStyleMove】
;功能：设置电机转动方式
;入口参数：开关量
;出口参数：StyleMove=电机转动方式
;影响寄存器：CX,SI,DI
SetStyleMove PROC
    PUSH AX
    PUSH DX
    MOV DX,PORT8255_C
    IN AL,DX                ;读取 8255C 端口
    SHR AL,1
    SHR AL,1                ;逻辑右移两位
    AND AL,3                ;使 PC3PC2 有效即得到输入的开关量
    MOV StyleMove,AL        ;开关量存入表示电机转动方式的变量 StyleMove
    POP DX
    POP AX
```

```
    RET
SetStyleMove ENDP
```

9.4 LED_DISPLAY

```
;子程序名：[LED_DISPLAY]
;功能：数码管显示子程序
;入口参数：  BITCODE=位码值
;            SEGTAB=段码值
;            BUFFER=需显示数据缓存区
;影响寄存器：AX,CX,DX,SI
;调用子程序：DELAY
LED_DISPLAY PROC NEAR
    PUSH AX
    PUSH CX
    PUSH DX
    PUSH SI                         ;保护现场
    LEA   BX,SEGTAB                   ;BX 为段码表首址
    MOV BITCODE,0FEH                  ;
    MOV   SI, 0                     ;SI 用作缓冲区指针，初值为 0
    MOV   CX, 8                      ;CX 用作循环计数器，初值为 8
ONE:
    MOV   AL, 0FFH                   ;位码=0FFH
    MOV   DX, PORT8255_B
    OUT   DX, AL                     ;熄灭所有数码管
    MOV   AL, BUFFER[SI]             ;从缓存区得到待查元素在表中的序号
    XLAT                            ;查表转换
    MOV   DX, PORT8255_A
    OUT   DX, AL                     ;送出一位信息的字形码（段码）
    MOV   AL,BITCODE                   ;位码
    MOV   DX, PORT8255_B
    OUT   DX, AL                     ;送出位码，选中某位数码管显示段码
    ROL   BITCODE, 1                 ;循环左移有效“0”--->形成下一个位码
    INC    SI                       ;修改输出缓冲区指针
    CALL DELAY0                        ;进行适当的延时使多位数码管同时显示
    LOOP    ONE                      ;循环，显示下一位信息
    POP SI                          ;恢复现场
    POP DX
    POP CX
    POP AX
    RET
LED_DISPLAY ENDP
```

9.5 SCANKEY

```
;子程序名：【SCANKEY】
;功能：列扫描按键得到键值
;出口参数：AL=键值
;影响寄存器：CX,DX
;调用子程序：TRANSLATE    \      DELAY2
SCANKEY PROC NEAR
    PUSH CX
    PUSH DX                ;保护寄存器
    MOV DX,PORT8255_B;B【列线】
    MOV AL,0
    OUT DX,AL              ;列输出全 0
    MOV DX,PORT8255_C;C【行线】
    IN AL,DX               ;读取 C 端口（行端口）
    AND AL,3
    CMP AL,03H             ;检测行信息(2 行->03H)是否全为 1，判断有无按键
    JZ NO_KEY              ;无按键时，转移后返回-1
    CALL DELAY2                ;调用延时子程序延时消抖
    IN AL,DX               ;读取行端口
    AND AL,03H
    CMP AL,03H             ;检测行信息是否全为 1，判断有无按键
    JZ NO_KEY              ;无按键时，转移后返回-1
    MOV AH,0FEH                ;指定列码，从 PB0 列开始扫描
    MOV CX,8               ;最多扫描 8 列
NEXT:MOV AL,AH                 ;列码->AL
    ROL AH,1               ;形成下一列的列码，为扫描下一列做准备
    MOV DX,PORT8255_B
    OUT DX,AL              ;输出列码
    MOV DX,PORT8255_C
    IN AL,DX               ;读取行码
    AND AL,03H
    CMP AL,03H             ;ZF=0  或  CX=0 则退出循环
    LOOPZ NEXT
    JZ NO_KEY              ;ZF=1,无按键，转移后返回-1
    ROR AH,1               ;存放形成的列行码->AX（恢复刚才的列码）
    CALL TRANSLATE             ;列行码转换为键值->AL
    JMP EXIT
NO_KEY:
    MOV AL,-1               ;没有按键 AL=-1
EXIT:
    CALL LED_DISPLAY      ;显示 LED（防止显示乱码）
    POP DX
    POP CX
```

```
    RET
SCANKEY ENDP
```

9.6 GetBuffer65

```
;子程序名：Getbuffer65
;功能：设置步进电机的步数/速度
;入口参数：十位 buffer+6，个位 buffer+5
;出口参数：步数/速度值 SetCount
;影响寄存器：AX,BX
Getbuffer65 PROC
    PUSH AX
    PUSH BX                          ;保护现场
    MOV AL,buffer+6                  ;转动步数送入 SetCount
    MOV BX,10
    MUL BL
    ADD AL,buffer+5
    MOV SetCount,AL
    POP BX                           ;恢复现场
    POP AX
    RET
Getbuffer65 ENDP
```

9.7 SetJiShu

```
;子程序名：SetJiShu
;功能：设置 8253 计数值得到不同的频率
;入口参数：SetCount
;出口参数：JuShu，SpeedNumber
;影响寄存器：AX,BX
SetJiShu PROC NEAR
    PUSH AX              ;保护现场
    PUSH BX
    INC SetCount         ;输入 0 对应 1 级速度，以此类推
    MOV AL,SetCount      ;保存速度级数到 SpeedNumber
    MOV SpeedNumber,AL
    MOV AL,101           ;根据速度级数通过 5000+100*(101-级数)计算 8253 计数值
    SUB AL,SetCount
    MOV BL,100
    MUL BL
    ADD AX,5000
    MOV JiShu,AX         ;将计数值传给 JiShu
    POP BX               ;恢复现场
```

```
    POP AX
    RET
SetJiShu ENDP
```

9.8 StepCountSet

```
子程序名：StepCountSet
;功能：把 buffer0-3（当前步数值）送入 StepCount
;入口参数：buffer0-3
;出口参数：StepCount
;影响寄存器：AX,BX
StepCountSet PROC NEAR
    PUSH AX
    PUSH BX                          ;保护现场
    MOV AL,buffer+3                  ;（buffer+3）-->AL
    MOV BX,10
    MUL BL
    ADD AL,buffer+2                  ;AL*10+（buffer+2）-->AL
    MUL BL
    ADD AL,buffer+1                  ;AL*10+（buffer+1）-->AL
    ADC AH,0
    MUL BX
    ADD AL,buffer                    ;AL*10+（buffer）-->AL
    ADC AH,0
    MOV StepCount,AX                 ;转动步数送入 StepCount
    POP BX                           ;恢复现场
    POP AX
    RET
StepCountSet ENDP
```

9.9 DELAY 系列

```
;子程序名：DELAY0
;功能：延时子程序：延时使多位数码管同时显示 F LED
;影响寄存器：CX
DELAY0 PROC NEAR
    PUSH CX
    MOV CX,500
        LOOP $
    POP CX
    RET
DELAY0 ENDP
;-----------------------------------------------------------------
```

```
;子程序名：DELAY1
;功能：延时子程序：延时用于防止输入数字时过快 主程序设置 0~9 之后
;影响寄存器：CX
DELAY1 PROC NEAR
    PUSH CX
    MOV CX,25000
    LOOP $
    POP CX
    RET
DELAY1 ENDP
;-------------------------------------------------------------------
;子程序名：DELAY2
;功能：延时子程序：用于扫描按键时按键消抖
;影响寄存器：CX
DELAY2 PROC NEAR
    PUSH CX
    MOV CX,10
    LOOP $
    POP CX
    RET
DELAY2 ENDP
```

9.10 HltFunction

```
;子程序名：[HltFunction]
;功能：数码管显示全 F 子程序
;影响寄存器：AX,DX
;调用子程序：DELAY0
HltFunction PROC NEAR
    PUSH AX
    PUSH DX                         ;保护现场
    MOV   AL, 0FFH                  ;位码=0FFH
    MOV   DX, PORT8255_B
    OUT   DX, AL                    ;熄灭所有数码管
    MOV   AL,8EH                    ;F 的段码
    MOV   DX, PORT8255_A
    OUT   DX, AL                    ;送出段码
    MOV   AL,0                      ;所有位都显示
    MOV   DX, PORT8255_B
    OUT   DX, AL                    ;送出位码
    CALL DELAY0                     ;进行适当的延时
    POP DX                          ;恢复现场
    POP AX
    RET
```

```
HltFunction ENDP
```

9.11 StepControlSet

```
;子程序名：StepControlSet
;功能：判断电机转动方式，并送下一次送入 StepControl 的值
;入口参数：ControlStyle 转动方式数组，以及偏移量 DI（StyleMove）
;出口参数：StepControl
;影响寄存器：BX,AX
;问题修改：顺时针-->右移,逆时针-->左移
StepControlSet PROC NEAR
    PUSH BX
    PUSH AX
    XOR BX,BX
    MOV BL,StyleMove            ;将偏移量送入 StyleMove
    MOV AL,[ControlStyle+BX]    ;将转动方式的相位值送给 StepControl
    MOV StepControl,AL
    CMP BX,2                    ;是否是单双 8 拍的转动方式
    JNB DanShuang8              ;如果是单双 8 拍，则跳转到 DanShuang8
    CMP bClockwise,1            ;否则，判断电机转动方向
    JZ  ControlClockwise        ;如果是正转，跳转到 ControlClockwise
ControlAntiClockwise:           ;如果是反转，顺势左移 1 位
    ROL [ControlStyle+BX],1
    JMP SCNEXT
ControlClockwise:               ;如果是正转，顺势右移 1 位
    ROR [ControlStyle+BX],1
    JMP SCNEXT
DanShuang8:                     ;单双 8 拍的转动方式
    CMP bClockwise,1            ;判断电机转动方向
    JZ DSClockwise              ;如果是正转，跳转到 DSClockwise
DSAntiClockwise:                ;如果是反转，选择数组下一个值送入 StepControl
    INC BX
    CMP BX,10                   ;相位值循环(偏移量 DI 增加)
    JNZ SCNEXT
    MOV BX,9
    JNZ SCNEXT
DSClockwise:                    ;如果是正转，选择数组上一个值送入 StepControl
    DEC BX
    CMP BX,1                    ;相位值循环(偏移量 DI 减少)
    JNZ SCNEXT
    MOV BX,2
    JNZ SCNEXT
SCNEXT:
    MOV StyleMove,BL            ;保存此时的偏移量值进 StyleMove
```

```
    MOV AL,[ControlStyle+BX]
    MOV StepControl,AL          ;将下一步相位值给到 StepControl
    POP AX
    POP BX
    RET
StepControlSet ENDP
```

9.12 AlterStep

```
;子程序名：AlterStep
;功能：步进电机的步数调整
;入口参数：步数（buffer 低 4 位）数组首地址 BX，步数位数 CX
;出口参数：buffer 低 4 位
;影响寄存器：BX,CX
;问题修改：处理特殊数字跳转及设置 9999 反转
AlterStep PROC
    PUSH CX                         ;保护现场
    PUSH BX
    CALL StepCountSet               ;获取显示屏当前步数
    MOV CX,4                        ;设置步数位数
    LEA BX,buffer                   ;数组首地址
    CMP bClockwise,1                ;判断正/反转
    JZ ClockwiseSet
AntiClockwiseSet:                   ;反转设置
    CMP BYTE PTR[BX+4],11H          ;判断步数是正是负
    JZ AddStep                      ;步数为负，+1
    CMP StepCount,0                 ;步数为 0.0000
    JNZ AntiClockwiseSet1
    MOV BYTE PTR[BX+4],11H
    INC BYTE PTR[BX]                ;则步数变为-.0001
    JMP AlterStep1
AntiClockwiseSet1:
    JMP SubStep                     ;步数为正，-1
ClockwiseSet:                       ;正转设置
    CMP BYTE PTR[BX+4],10H          ;判断步数是正是负
    JZ AddStep                      ;步数为正，+1
    CMP StepCount,1                 ;步数是否为-.0001
    JNZ ClockwiseSet1
    MOV BYTE PTR[BX+4],10H
    DEC BYTE PTR[BX]                ;步数变为 0.0000
    JMP AlterStep1
ClockwiseSet1:
    JMP SubStep                     ;步数为负，-1
AddStep:
```

```
    INC BYTE PTR [BX]                    ;步数+1
    CMP BYTE PTR [BX],0AH                ;低位是否产生进位
    JNZ AlterStep1                       ;无进位，则退出
    MOV BYTE PTR [BX],0                  ;有进位，处理进位
    INC BX
    LOOP AddStep
    SUB BX,4
    MOV CX,4                             ;四位都有进位则达到最大值
AddStep1:                                ;设置最大值 0.9999
    MOV BYTE PTR [BX],9
    INC BX
    LOOP AddStep1
    XOR bClockwise,1          ;步数>9999,使电机反转
    CMP BUFFER+7,0BH          ;同时修改转动方式
    JZ BD_fanzhuan
    CMP BUFFER+7,0DH
    JZ BD_fanzhuan
    DEC BUFFER+7
    JMP AlterStep1
BD_fanzhuan:
    INC BUFFER+7
    JMP AlterStep1
SubStep:
    DEC BYTE PTR [BX]         ;步数-1
    CMP BYTE PTR [BX],0FFH    ;低位是否产生借位
    JNZ AlterStep1            ;无借位，则退出
    MOV BYTE PTR [BX],9       ;有借位，处理借位
    INC BX
    LOOP SubStep
AlterStep1:
    POP BX                    ;恢复现场
    POP CX
    RET
AlterStep ENDP
```

9.13 TIMERO

```
;子程序名：TIMERO
;功能：中断子程序
;影响变量：StepDec,StepControl,bNeedDisplay
;影响寄存器：AX,DX
TIMERO PROC NEAR
    PUSH AX
    PUSH DX
```

```
    MOV AL,StepControl          ;将下一次要送给步进电机的值送给 AL
    MOV DX,PORT8255_C           ;将 StepControl 通过 8255C 口输出
    OUT DX,AL
    CALL StepControlSet         ;设置下一步相位值
    MOV bNeedDisplay,1          ;设置需要显示新步数
    CALL AlterStep              ;调用步数调整子程序对显示屏显示的步数进行调整
    CMP StepDec,-1              ;判断是否设置步数
    JZ TIMERO_1                 ;没有设置步数即 B,C 模式下，直接发结束中断 EOI
    DEC StepDec                 ;设置步数，中断一次步数-1
    CMP StepDec,0               ;判断 D,E 功能下设置的步数 StepDec 有没有走完
    JNZ TIMERO_1                ;如果走完了需要关中断停止，并使 bFirst 为 1
    MOV DONE,1                  ;中断完成，标志位置'1'
    CLI                         ;关中断
    MOV bFIRST,1                ;电机-->停机状态
TIMERO_1:
    MOV DX,PORT8259_0           ; 发结束中断 EOI
    MOV AL,20H
    OUT DX,AL
    POP DX
    POP AX
    IRET
TIMERO ENDP
```

9.14 SEND8251

```
;子程序名：SEND8251
;功能：发送 BUFFERsend 初始数据
;影响寄存器：DI,AX,DX,
SEND8251 PROC
    PUSH DI
    PUSH AX
    PUSH DX
    LEA DI,BUFFERsend           ;设置发送数据快地址指针
    MOV CX,8                    ;设置计数器初值
    ;------发送数据----------
SEND:
    MOV DX,PORT8251_1           ;
    IN AL,DX                    ;
    AND AL,01H                  ;查询 TxRDY 有效否？
    JZ SEND                     ;TxRDY=0，无效则等待
    MOV DX,PORT8251_0           ;
    MOV AL,[DI]                 ;向 8251 输出 1 个字节数据
    OUT DX,AL                   ;
    INC DI                      ;修改地址指针
```

```
    LOOP SEND                    ;循环
    POP DX
    POP AX
    POP DI
    RET
SEND8251 ENDP
```

9.15 RECEUVE8251

```
;子程序名：RECEIVE8251
;功能：接收初始数据子程序
;影响变量：BUFFER
;影响寄存器：BX,AX,DX
RECEIVE8251 PROC
    PUSH BX
    PUSH AX
    PUSH DX
    LEA BX,BUFFER                ;设置接收数据块地址指针
    MOV CX,8                     ;设置计数器初值
    ;------接收数据----------
RECEIVE:
    MOV DX,PORT8251_1            ;
    IN AL,DX                     ;
    TEST AL,02H                  ;查询 RxRDY 有效否？
    JZ RECEIVE                   ;RxRDY=0，无效则等待
    MOV DX,PORT8251_0            ;
    IN AL,DX                     ;RxRDY=1，读入数据
    MOV [BX],AL                  ;保存数据
    INC BX                       ;修改地址指针
    LOOP RECEIVE                 ;循环
    POP DX
    POP AX
    POP BX
    RET
RECEIVE8251 ENDP
    END START
```

9.16 0809 从机思想

9.16.1 新增变量

```
STOPPINGTEST DB 0
```

9.16.2 主程序修改

```
START:
    MOV AX,@DATA
    MOV DS,AX                    ;装载 DS
    ;-------------初始化部分
    CALL INIT8253                ;8253 初始化
    CALL INIT8255                ;8255 初始化
    CALL INIT8259                ;8259 初始化
    CALL INIT8251                ;8251 初始化
    CALL INTERRUPT_VECTOR        ;中断向量表初始化
   ; CALL RECEIVE8251            ;调用子程序获取初始化数据
    CALL INITMOTOR               ;步进电机模块初始化
START_1:
    CALL LED_DISPLAY             ;调用数码管显示子程序

    CMP bFirst,1                 ;电机转动了一步，且此时没有按键按下，判断电机此时的状
态
    JZ START_2
    MOV CX,20
    CALL DELAY
    LOOP $
    STI
    MOV CX,20
    CALL DELAY
    LOOP $
   MOV DX,PORT0809               ;选择其中一个通道
    OUT DX,AL                    ;启动转换

    CALL DELAY                   ;延时

    IN AL,DX                     ;读取
    MOV CX,20
    CALL DELAY
    LOOP $
    CALL TRANSLATETEST           ;调用子程序转换成数字信号
    JMP START_1
START_2:
CLI
    CALL RECEIVE8251             ;调用子程序获取初始化数据,设置 buffer
    CALL Getbuffer65             ;设置 SetCount
    CMP buffer+7,0BH             ;如果是功能 B
    JZ BC                        ;跳转到 BC，设置步速
    CMP buffer+7,0CH             ;如果是功能 C
```

```
    JZ BC                          ;跳转到 BC，设置步速
DE:                                ;功能 DE，设置步数
    CMP BUFFER+7,0DH
    JZ DTEST
    MOV BCLOCKWISE,0
    JMP DE1
DTEST:
    MOV BClOCKWISE,1
DE1:
    MOV AL,SetCount                ;AL <-- SetCount
    MOV BL,10                      ;BL <-- 10
    MUL BL                         ;设置步数为 10*SetCount
    MOV StepDec,AX                 ;再将步数送给 StepDec(即输入 99，设置步数为 990
步)
    CMP StepDec,0                  ;若初始设置为 0
    JZ  START_3                    ;跳转到开始继续等待按键
    JMP START_1
START_3:
    MOV BFIRST,1
    JMP START_1                    ;否则开中断
BC:                                ;功能 BC，设置步速
    CMP BUFFER+7,0BH
    JZ BTEST
    MOV BCLOCKWISE,0
    JMP BC1
BTEST:
    MOV BClOCKWISE,1
BC1:
    MOV StepDec,-1                 ;清除步数
    CALL SetJiShu                  ;调用子程序设置 8253 计数值
    CALL INIT8253                  ;设置新的速度
    JMP START_1
```

9.16.3 新增子程序

```
RANSLATETEST PROC NEAR
    PUSH CX
    MOV AH,AL                      ;
    AND AL,0FH                     ;将 AL 中高四位清零
    AND AH,0F0H                    ;将 AH 中低四位清零
    MOV CL,4                       ;将 4 存入 CL 中
    SHR AH,CL                      ;将 AH 右移 4 位
    MOV STOPPINGTEST,AH
    CMP STOPPINGTEST,0FH
```

```
        JNZ TESTNEXT
        MOV STOPPINGTEST,AL
        CMP STOPPINGTEST,0FH
        JNZ TESTNEXT
        MOV BFIRST,1
        CLI
        TESTNEXT:

        POP CX
        RET
TRANSLATETEST ENDP


DELAY PROC NEAR
        PUSH CX
        MOV CX,0100H
L1: LOOP L1
        POP CX
        RET
DELAY ENDP
```