

微机原理与接口技术

课程设计报告



题 目：共享单车智能动态密码锁

学 校： 上海海事大学

报 告 人： 徐家晨

学 号： 201510230194

班 级： 电气自动化 153

指导教师： 王天真

摘要

微型计算机原理控制应用技术飞速发展，纵观我们现在生活的各个领域，从神州飞船，到手腕上的智能手表，从计算机的网络通讯与数据传输，到电力电子器件的控制，这些都离不开微型计算机原理控制。本设计为一个基于 STAR ES598PCI 8086 实验系统的智能密码锁，采用“飞思卡尔”的 32bit 微控制器 MK60DN512ZVLL10 单片机作为密码锁的控制器，利用 74HC08 芯片作为脉冲转换模块，通过创新的硬件结构及软件设计实现了控制器控制智能密码锁定时自动更换密码的功能。

目录

第一章	总体方案设计.....	1
1.1	课程设计任务.....	1
1.2	总体设计方案.....	1
1.2.1	硬件	1
1.2.2	软件	1
第二章	硬件设计方案.....	1
2.1	8086 实验板部分	1
2.1.1	电路原理图	1
2.1.2	片选说明	5
2.1.3	接线说明	5
2.2	外置控制器部分.....	6
2.2.1	原理图	6
第三章	软件设计方案.....	8
3.1	主程序流程图.....	8
3.2	中断服务程序流程图.....	9
3.3	控制器部分程序流程图.....	9
3.4	程序清单及注释.....	10
3.4.1	密码锁端	10
3.4.2	控制器端	27
第四章	实验结果及分析.....	29
4.1	实验结果.....	29
4.2	分析.....	29
第五章	课设体会和总结.....	30
第六章	参考文献.....	31

第一章 总体方案设计

1.1 课程设计任务

综合使用 8255、8253、8259 等芯片，设计一种共享单车动态密码锁，进行硬件搭建及软件编写，密码锁按键输入及密码比对、控制器定时控制密码库定时动态改变、控制器显示屏模拟手机屏幕显示当前密码及改变时间倒数计时、点阵显示屏显示开锁状态等功能。

1.2 总体设计方案

1.2.1 硬件

利用 8279 芯片实现键盘输入及数码管显示，利用 8253 芯片结合蜂鸣器实现不同音阶音频输出，利用 8259 芯片实现控制器触发 8086 中断更换密码库，利用 8155、8255 实现 16*16 点阵显示器显示开锁与闭锁标志。利用“飞思卡尔”的 32bit 微控制器 MK60DN512ZVLL10 单片机作为密码锁的控制器，定时产生脉冲控制信号，利用 74HC08 芯片作为脉冲转换模块，产生 8259 芯片的中断触发信号。利用控制器上的 OLED 显示屏显示当前密码及改变时间倒数计时。

1.2.2 软件

本设计要求软件部分实现各芯片的正确初始化，完成各程序段的编写，实现键盘输入及数码管显示密码，检测密码输入状态，在密码输入完毕之后进行密码比对，完成中断服务函数的编写，实现中断触发之后更改内存中的预设密码。完成控制器模块代码编写，实现定时产生控制脉冲、当前密码显示及密码改变时间倒数计时。

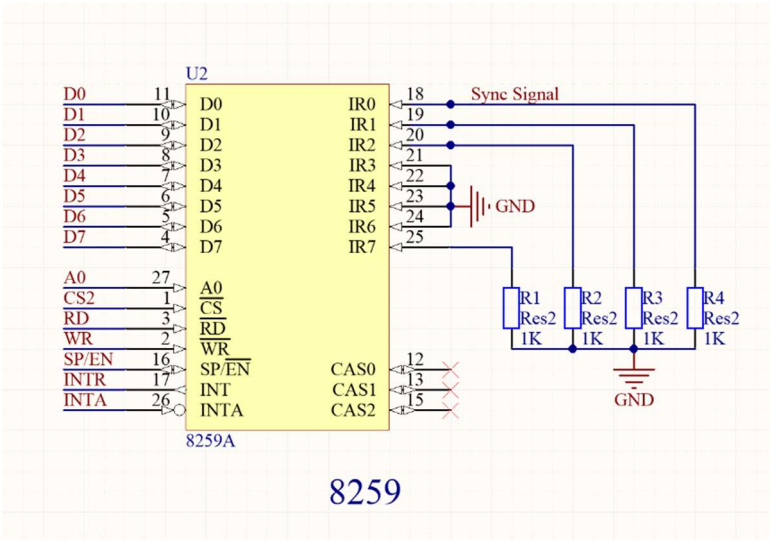
第二章 硬件设计方案

2.1 8086 实验板部分

2.1.1 电路原理图

1. 中断模块

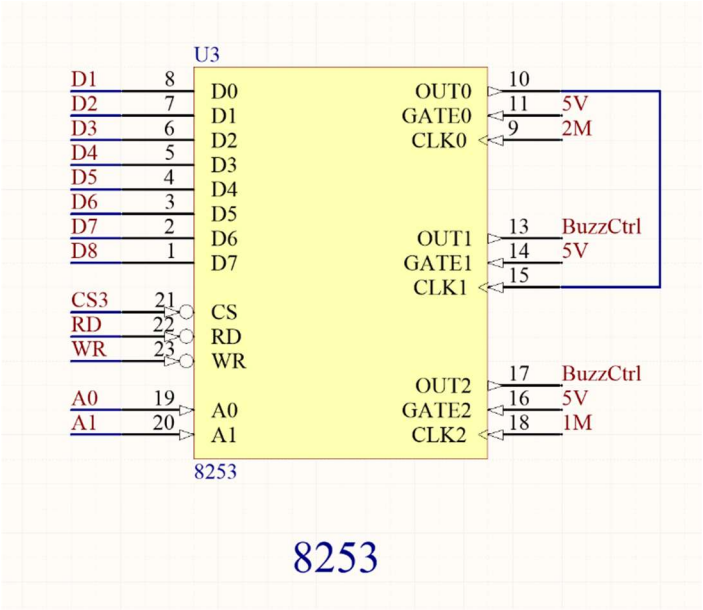
使用 8259A 芯片 IR0 作为脉冲输入端，触发中断，用于实现密码更换。



8259及其外围电路

2. 音频模块

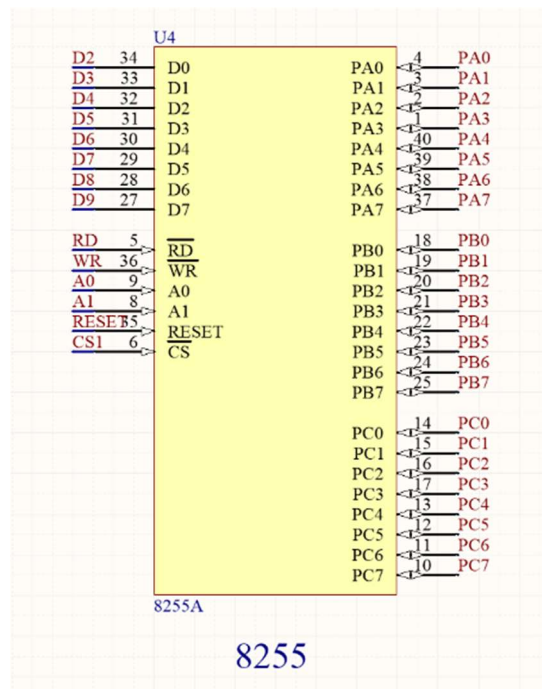
使用 8253 产生不同频率方波输出给蜂鸣器产生不同声音。其中计数器 0 工作在方式 2，以 2M 脉冲信号作为输入，实现晶振信号一次分频；计数器 1 工作在方式 3，以计数器 0 输出信号作为输入，产生方波信号控制蜂鸣器实现报警音；计数器 2 工作在方式 3，以 1M 脉冲信号作为输入，产生方波信号控制蜂鸣器实现按键音。



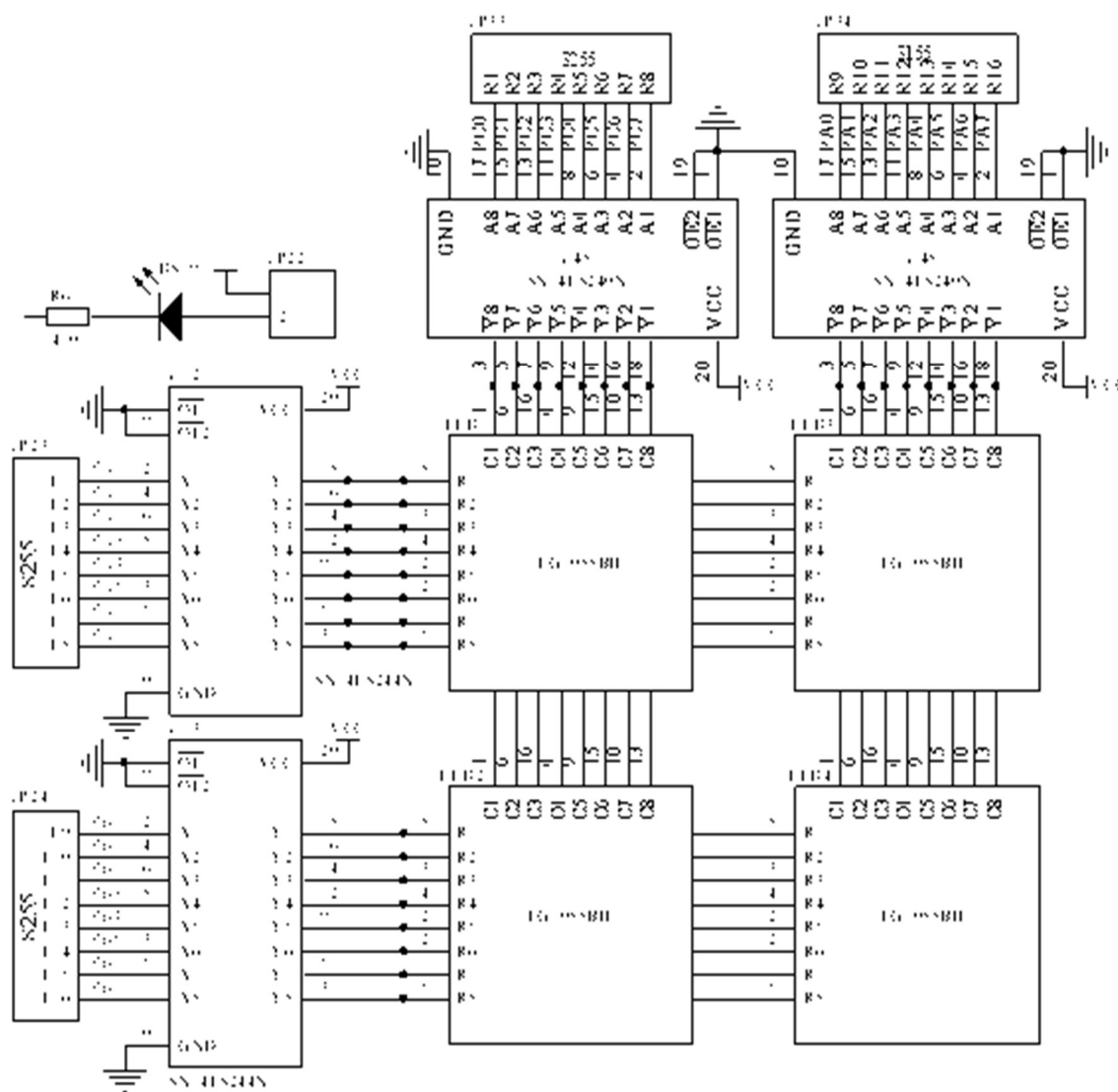
8253及其外围电路

3. 点阵显示模块

8255 及 8155 用于控制 16*16 LED 点阵显示器显示智能锁锁定状态。



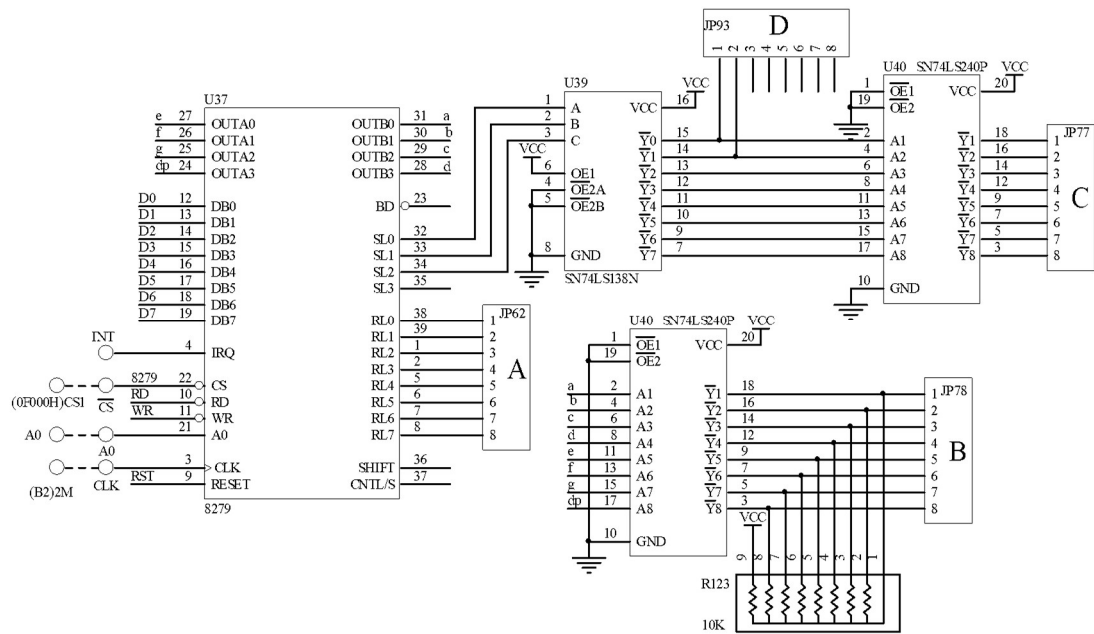
8255及其外围电路



LED 点阵、8255、8155

4. 输入模块

使用 8279 及 8255 实现矩阵键盘的输入与密码在数码管的实时显示。



8279、矩阵键盘、数码管电路

2. 1. 2 片选说明

芯片	片选号
8255	CS1
8259	CS2
8253	CS3
8155	CS4

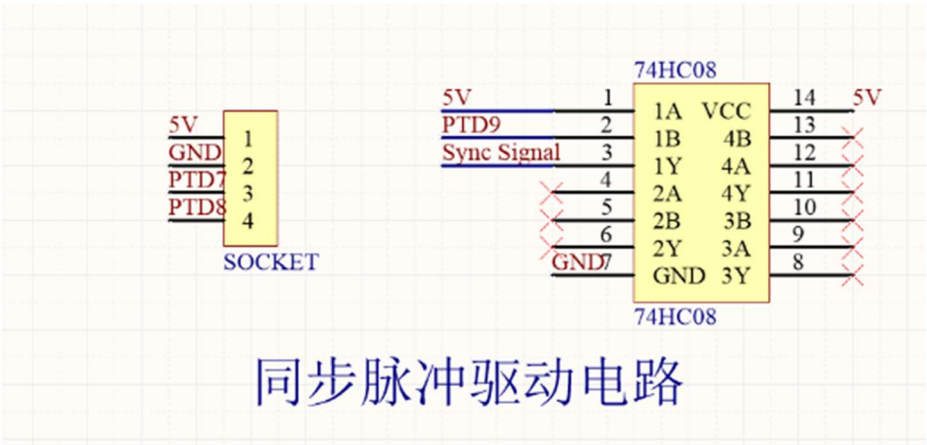
2. 1. 3 接线说明

区域及引脚	对应区域及引脚
E5 区 CS, A0	A3 区 CS5, A0
E5 区 CLK	B2 区 2M
E5 区 A、B、C、D	G5 区 A、B、C、D
B3 区 CS A0	A3 区 CS2 A0
B3 区 INT, INTA	ES8688 INTR INTA
B3 区 IR0	同步信号

区域及引脚	对应区域及引脚
B4 区 CS, A0, A1	A3 区 CS1, A0, A1
B4 区 JP56	G6 区 JP65
C5 区 CS, A0, A1	A3 区 CS3, A0, A1
C5 区 CLK0	B2 区 2M
C5 区 OUT0	C5 区 CLK1
C5 区 OUT1, OUT2	D1 区 Ctrl
C5 区 GATE0, GATE1, GATE2	C1 区 VCC
C5 区 CLK2	B2 区 1M
B4 区 CS, IO/M	A3 区 CS4, A8
B4 区 JP56, JP53	A2 区 JP23, JP24
B4 区 JP52, JP76	A2 区 JP33, JP34

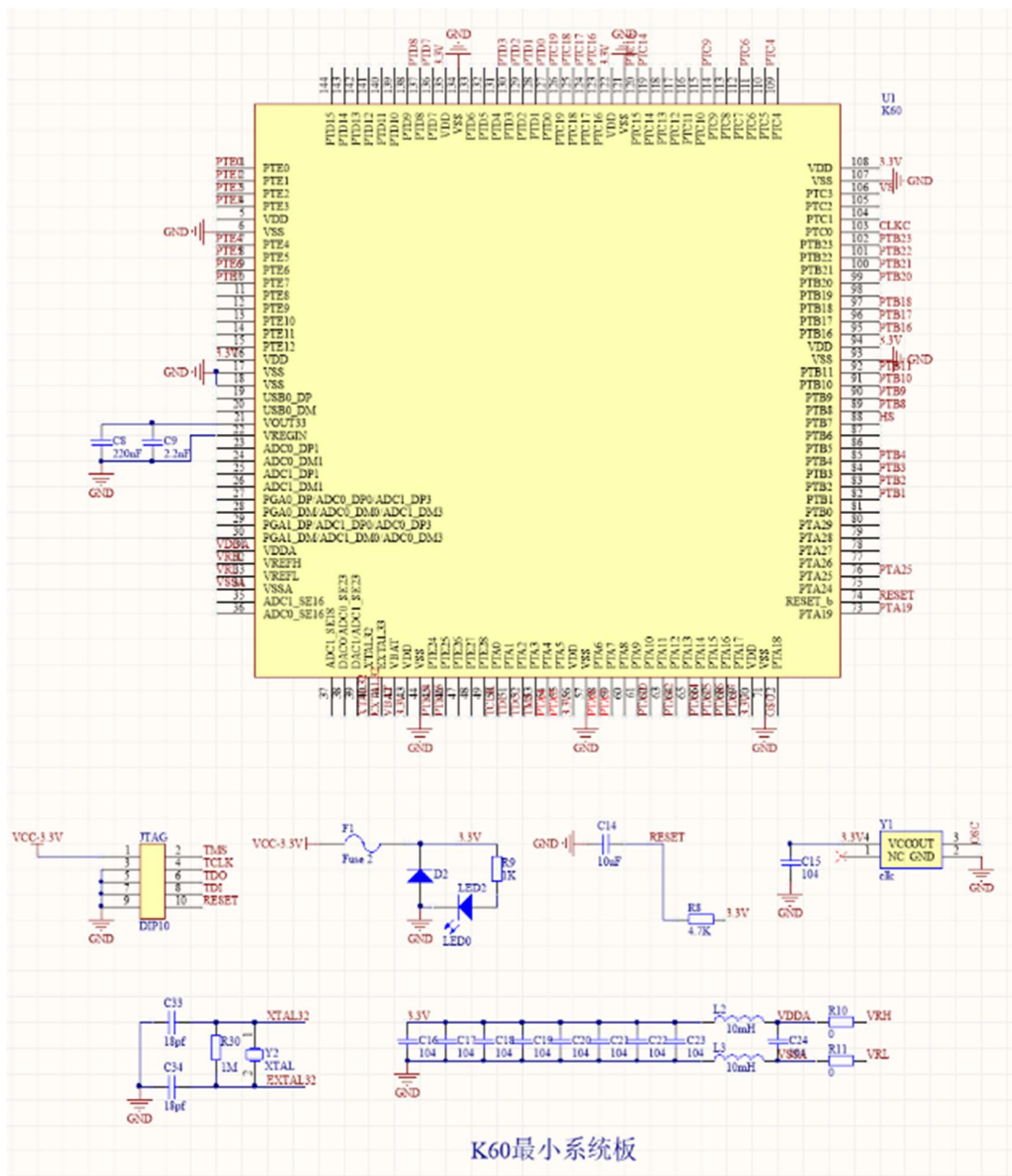
2.2 外置控制器部分

2.2.1 原理图



同步脉冲驱动电路

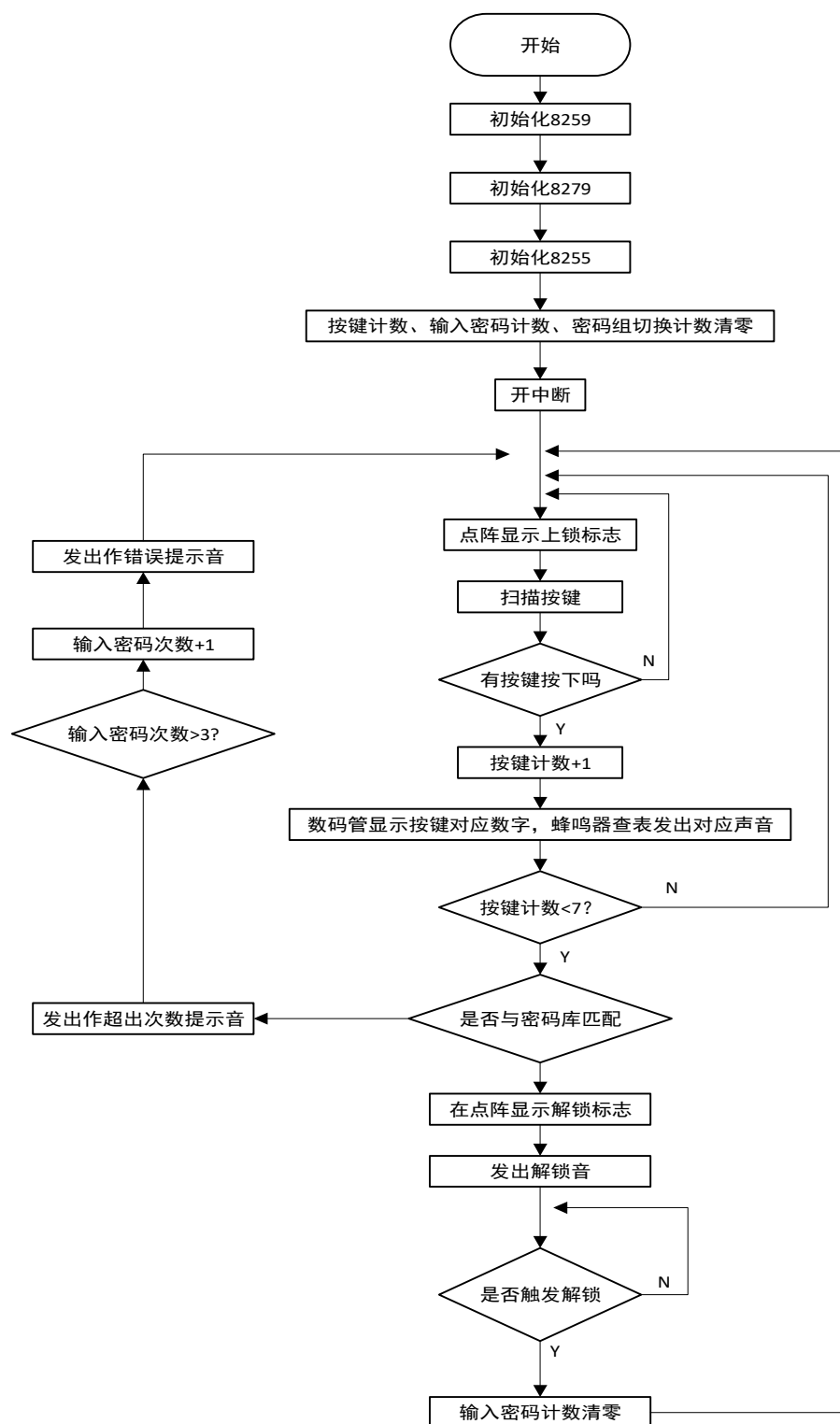
同步脉冲驱动电路使用一片 74HC08 与门芯片将 3 单片机 3.3V 脉冲信号转换为可以触发 8259 的 5V 脉冲信号。



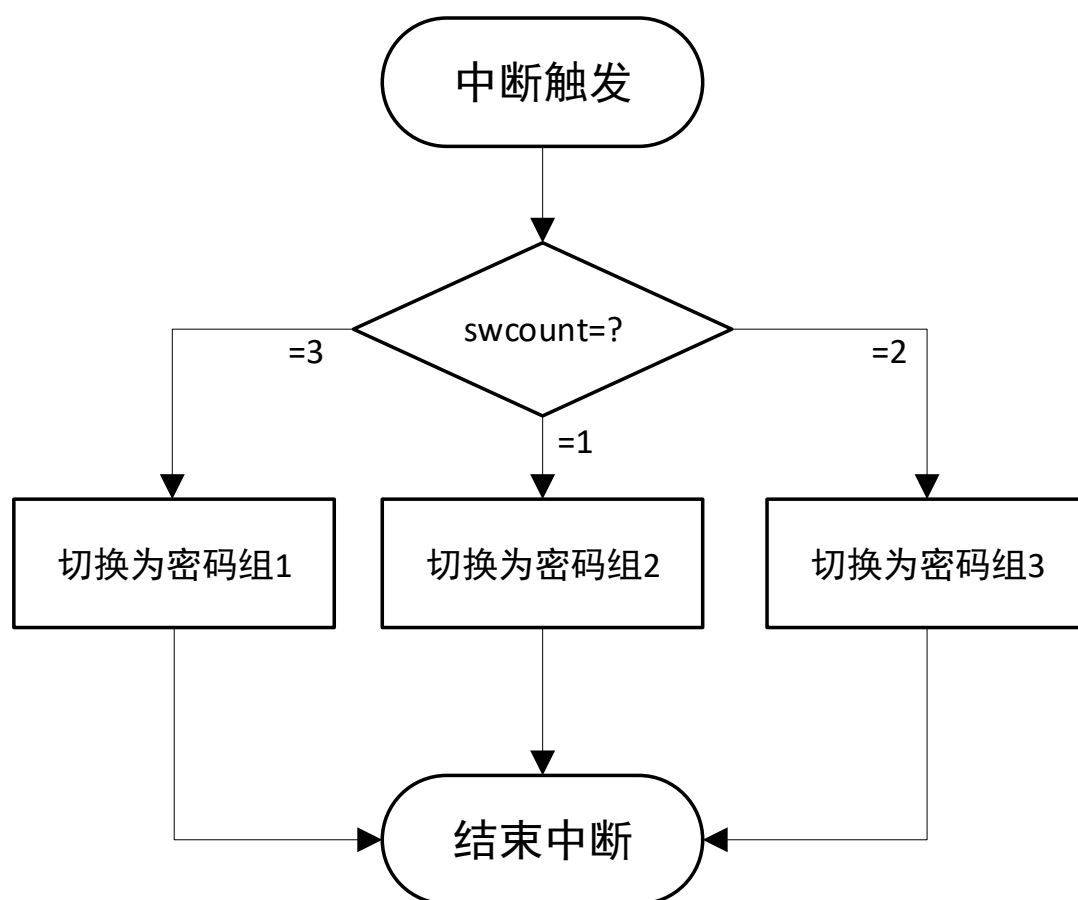
MK60DN512ZVLL10单片机及其外围电路

第三章 软件设计方案

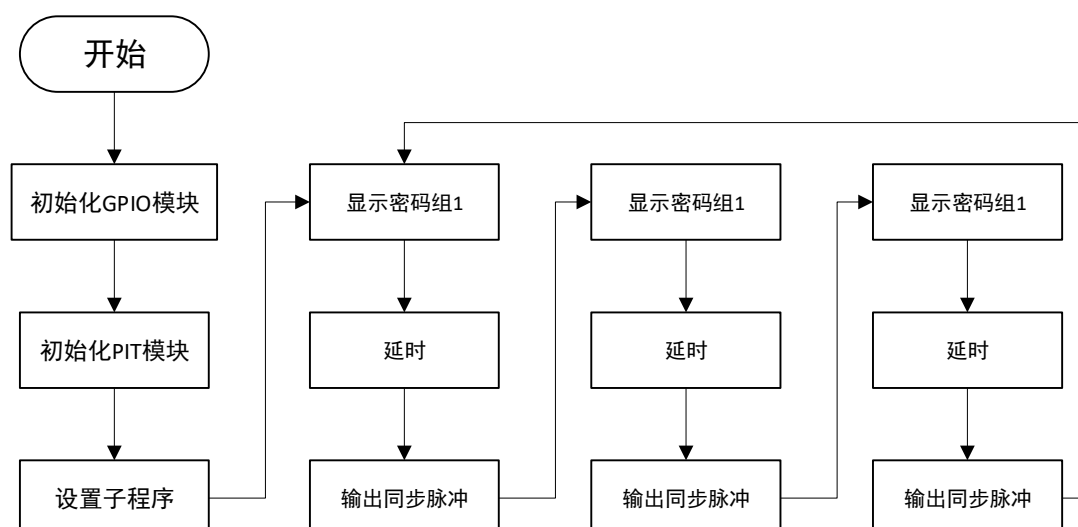
3.1 主程序流程图



3.2 中断服务程序流程图



3.3 控制器部分程序流程图



3.4 程序清单及注释

3.4.1 密码锁端

```
.MODEL TINY
CMD_8279 EQU 0BF01H ;8279 CS5
DATA_8279 EQU 0BF00H
COM_ADD55 EQU 0F003H ;8255 CS1
PA_ADD55 EQU 0F000H
PB_ADD55 EQU 0F001H
PC_ADD55 EQU 0F002H
IO8259_0 EQU 0E000H ;8259 CS2
IO8259_1 EQU 0E001H
ADDR_8155_PA EQU 0C101H ;8155 PA 口
ADDR_8155_C EQU 0C100H ;8155 控制口
LINE1 EQU PA_ADD55 ;行线 1
LINE2 EQU PB_ADD55 ;行线 2
ROW1 EQU PC_ADD55 ;列线 1
ROW2 EQU ADDR_8155_PA;列线 2
IO8259_1 EQU 0E001H
COM_ADDR EQU 0D003H ;8253 CS3
T0_ADDR EQU 0D000H
T1_ADDR EQU 0D001H
T2_ADDR EQU 0D002H

.STACK 100
.DATA
;显示器取模
LOCKLOGO DB 00H,00H,0FH,00H,10H,80H,10H,80H,20H,40H,20H,40H,20H,40H,7FH,0E0H
DB 7FH,0E0H,7FH,0E0H,7FH,0E0H,7FH,0E0H,7FH,0E0H,7FH,0E0H,7FH,0E0H,7FH,0E0H
UNLOCKLOGO DB 00H,00H,00H,1CH,00H,22H,00H,22H,00H,41H,00H,41H,00H,41H,7FH,0E0H
DB 7FH,0E0H,7FH,0E0H,7FH,0E0H,7FH,0E0H,7FH,0E0H,7FH,0E0H,7FH,0E0H,7FH,0E0H
NONE DB 00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
DB 00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
KEYCOUNT DB ?
PWCOUNT DB ?
SWCOUNT DB ?
LED_TAB DB 0C0H,0F9H,0A4H,0B0H,99H,92H,82H,0F8H
DB 080H,90H,88H,83H,0C6H,0A1H,86H,8EH
PW0 DB 01H,02H,03H,04H,05H,06H ;初始密码
```

```

PW          DB  6 DUP(?)          ;记录密码
PW1         DB  01H,02H,03H,03H,02H,01H
PW2         DB  06H,05H,04H,03H,02H,01H
PW3         DB  01H,02H,03H,04H,05H,06H
NUM         DB  00H              ;记录密码错误次数，起始为零
LOCK_FLAG   DB  01H
LED_DATA    DB  01111111B        ;密码正确亮绿灯
DB  10111111B                    ;待机状态亮黄灯
DB  11011111B                    ;密码错误亮红灯
FQ_H        DB  019H              ;音调频率，高位
DB  017H
DB  015H
DB  014H
DB  012H
DB  011H
DB  012H
DB  09H
DB  08H
DB  07H
DB  07H
DB  06H
DB  05H
DB  05H
FQ_L        DB  012H              ;音调频率，低位
DB  03H
DB  017H
DB  032H
DB  077H
DB  036H
DB  013H
DB  056H
DB  051H
DB  058H
DB  016H
DB  038H
DB  068H
DB  006H
.CODE
START:      MOV  AX,@DATA
MOV  DS,AX
    
```

```
MOV     ES,AX;8259
NOP
CALL Init8259;8259 初始化
CALL WriIntver;8259 定义中断向量
CALL INIT_IO;点阵显示初始化
CALL INIT8279 ;初始化键盘子程序
CALL INIT8255 ;初始化 8255 程序
MOV  KEYCOUNT,0;按键计数
MOV  PWCOUNT,0
MOV  SWCOUNT,0;切换计数
MOV  NUM,0;输入的密码次数计数
CALL CLEARDISP
STI  ;开中断
START1:
XCHG CL,LOCK_FLAG
PUSH CX
CMP  CL,00H
JNZ  DISP_LOCK
START3:
CALL SCAN_KEY ;键扫描
JNC  START1 ;没有按键
XCHG AL,KEYCOUNT
INC  AL
CMP  AL,7
JNZ  START2
MOV  KEYCOUNT,0
CALL INIT8279_1; 8 个数码块全有字符显示后,再按键,清除显示
JMP  START1
START2:
XCHG AL,KEYCOUNT
MOV  CX,0000H
MOV  CX,WORD PTR KEYCOUNT
SUB  CX,0001H
MOV  DI,OFFSET PW
ADD  DI,CX
AND  AL,0FH
MOV  [DI], AL ;[DL]中存入按键
PUSH AX
PUSH CX
```



```
PUSH DX
MOV CL,AL
MOV DX,COM_ADDR      ;8253
MOV AL,10110111B
OUT DX,AL             ;计数器 T2 为模式 3 状态, 输出方波,BCD 码计数
MOV DX,T2_ADDR
LEA BX,FQ_L
MOV AL,CL
XLAT
OUT DX,AL
LEA BX,FQ_H
MOV AL,CL
XLAT
OUT DX,AL ; 发出按键所对应的声音
CALL DL500ms
MOV DX,COM_ADDR
MOV AL,10110111B
OUT DX,AL ;延时 0.5 秒后关闭按键所发出的声音
POP DX
POP CX
POP AX

CALL KEY_NUM ;键值转换为键号
LEA BX,LED_TAB ;字型码表
XLAT
CALL WRITE_DATA
CALL DL500MS
CMP CX,5H
JZ COMPARE1
JMP START1
START_EXIT: JMP $
DISP_LOCK PROC NEAR
POP CX
LEA SI,LOCKLOGO
CALL DISP_CH;
XCHG CL,LOCK_FLAG
STC
JMP START3
DISP_LOCK ENDP
DISP_UNLOCK PROC NEAR
```

```

LEA SI,UNLOCKLOGO
CALL DISP_CH;
POP CX;恢复 lockflag
XCHG CL,LOCK_FLAG;恢复 CX
STC
JMP LOCK12
DISP_UNLOCK ENDP
START11 PROC NEAR
;置锁标志位
PUSH AX
MOV AL,01H
XCHG AL,LOCK_FLAG
POP AX
JMP START1
START11 ENDP
COMPARE1 PROC NEAR
JMP COMPARE
COMPARE1 ENDP
LOCK1 PROC NEAR
LOCK11:

XCHG CL,LOCK_FLAG;保护 cx
PUSH CX;保护 lockflag
CMP CL,01H
JNZ DISP_UNLOCK
LOCK12:
CALL SCAN_KEY ;键扫描
JNC LOCK11 ;没有按键
PUSH AX
PUSH CX
PUSH DX
AND AL,0FH

CMP AL,0FH ;扫描键盘，直到按了 F 键锁上锁
JNZ LOCK1
MOV CL,AL
MOV DX,COM_ADDR
MOV AL,10110111B
OUT DX,AL ;计数器 T2 为模式 3 状态，输出方波,BCD 码计数

```

```
MOV     DX,T2_ADDR
MOV  AL,06H
OUT  DX,AL
MOV  AL,05H
OUT  DX,AL    ;音
CALL DL500ms
MOV  DX,COM_ADDR
MOV  AL,10110111B
OUT  DX,AL
MOV  DX,COM_ADDR
MOV  AL,10110111B
OUT  DX,AL    ;计数器 T2 为模式 3 状态，输出方波,BCD 码计数
MOV  DX,T2_ADDR
MOV  AL,068H
OUT  DX,AL
MOV  AL,05H
OUT  DX,AL    ;音
CALL DL500ms
MOV  DX,COM_ADDR
MOV  AL,10110111B
OUT  DX,AL
MOV  DX,COM_ADDR
MOV  AL,10110111B
OUT  DX,AL    ;计数器 T2 为模式 3 状态，输出方波,BCD 码计数
MOV  DX,T2_ADDR
MOV  AL,038H
OUT  DX,AL
MOV  AL,06H
OUT  DX,AL    ;音
CALL DL500ms
MOV  DX,COM_ADDR
MOV  AL,10110111B
OUT  DX,AL    ;以上为上锁音 si la so
MOV  NUM,0
POP  DX
POP  CX
POP  AX
JZ   START11
LOCK1  ENDP
COMPARE  PROC NEAR
```

```

LEA SI,PW0    ;输入的密码与原密码比较
LEA DI,PW
MOV CX,7 ;比对6次
CLD
REPE CMPSB
CMP CX,0
JZ RIGHT
JNZ ERROR
RET
COMPARE ENDP
RIGHT PROC NEAR ;密码正确
PUSH BX
PUSH DX
PUSH CX
;置解锁标志位
MOV CL,00H
XCHG CL,LOCK_FLAG

;MOV CX,4 ;闪烁四次
;LOOP1: MOV AL,0
;MOV DX,PA_ADD55
;LEA BX,LED_DATA
;XLAT
;OUT DX,AL ;亮绿灯
;CALL DL500ms
;MOV AX,0FFH
;OUT DX,AX
;CALL DL500ms
;LOOP LOOP1
CALL CLEAR
CALL DL500MS
MOV DX,COM_ADDR
MOV AL,10110111B
OUT DX,AL ;计数器 T2 为模式 3 状态，输出方波,BCD 码计数
MOV DX,T2_ADDR
MOV AL,038H
OUT DX,AL
MOV AL,06H
OUT DX,AL ;音

```

```
CALL DL500ms
MOV DX,COM_ADDR
MOV AL,10110111B
OUT DX,AL
MOV DX,COM_ADDR
MOV AL,10110111B
OUT DX,AL ;计数器 T2 为模式 3 状态,输出方波,BCD 码计数
MOV DX,T2_ADDR
MOV AL,068H
OUT DX,AL
MOV AL,05H
OUT DX,AL ;音
CALL DL500ms
MOV DX,COM_ADDR
MOV AL,10110111B
OUT DX,AL
MOV DX,COM_ADDR
MOV AL,10110111B
OUT DX,AL ;计数器 T2 为模式 3 状态,输出方波,BCD 码计数
MOV DX,T2_ADDR
MOV AL,06H
OUT DX,AL
MOV AL,05H
OUT DX,AL ;音
CALL DL500ms
MOV DX,COM_ADDR
MOV AL,10110111B
OUT DX,AL ;以上为开锁音 so la si
POP CX
POP BX
POP DX
JMP LOCK1
```

RIGHT ENDP

```
ERROR PROC NEAR ;密码错误
PUSH BX
PUSH DX
PUSH CX
; MOV CX,4
```

```
;LOOP2:  MOV  AL,2
;        MOV  DX,PA_ADD55
;        LEA  BX,LED_DATA
;        XLAT
;        OUT  DX,AL    ;亮红灯
;        CALL DL500ms
;        MOV  AX,0FFH
;        OUT  DX,AX
;        CALL DL500ms
;        LOOP LOOP2
CALL DL500MS
MOV  DX,COM_ADDR
MOV  AL,10110111B
OUT  DX,AL          ;计数器 T2 为模式 3 状态，输出方波,BCD 码计数
MOV  DX,T2_ADDR
MOV  AL,012H
OUT  DX,AL
MOV  AL,049H
OUT  DX,AL          ;音
CALL DL500MS
MOV  DX,COM_ADDR
MOV  AL,10110111B
OUT  DX,AL
MOV  DX,COM_ADDR
MOV  AL,10110111B
OUT  DX,AL          ;计数器 T2 为模式 3 状态，输出方波,BCD 码计数
MOV  DX,T2_ADDR
MOV  AL,030H
OUT  DX,AL
MOV  AL,049H
OUT  DX,AL          ;音
CALL DL1S
MOV  DX,COM_ADDR
MOV  AL,10110111B
OUT  DX,AL
CALL CLEAR
POP  BX
POP  DX
INC  NUM
```

```
CMP  NUM,3
JZ   ALARM
JMP  START1
ERROR ENDP

;报警
ALARM  PROC NEAR      ;8253
MOV  CX,2
LOOP3:  MOV  DX,COM_ADDR
MOV  AL,35H
OUT  DX,AL           ;计数器 T0 设置在模式 2 状态,BCD 码计数
MOV  DX,T0_ADDR
MOV  AL,00H
OUT  DX,AL
MOV  AL,10H
OUT  DX,AL           ;CLK0/1000
MOV  DX,COM_ADDR
MOV  AL,77H
OUT  DX,AL           ;计数器 T1 为模式 3 状态,输出方波,BCD 码计数
MOV  DX,T1_ADDR
MOV  AL,00H
OUT  DX,AL
MOV  AL,10H
OUT  DX,AL           ;CLK1/1000
CALL DL5S
MOV  DX,COM_ADDR
MOV  AL,77H
OUT  DX,AL

JMP  START
ALARM  ENDP

;8279 初始化
INIT8279 PROC NEAR
MOV  DX,CMD_8279      ;CMD_8279 为写命令地址、读状态地址
MOV  AL,34H           ;可编程时钟设置,设置分频系数(20 分频)
OUT  DX,AL
MOV  AL,0             ;8*8 字符显示,左边输入,外部译码键扫描方式
OUT  DX,AL
;   MOV  AL,0A0H
```

```
; OUT DX,AL
CALL INIT8279_1
RET
INIT8279 ENDP
INIT8279_1 PROC NEAR
CALL CLEAR ;清显示
MOV AL,90H ;从第一个数码管开始移位显示
OUT DX,AL
RET
INIT8279_1 ENDP
CLEAR PROC NEAR
MOV DX,CMD_8279
MOV AL,0DEH ;清除命令
OUT DX,AL
WAIT1: IN AL,DX
TEST AL,80H
JNZ WAIT1 ;显示RAM清除完毕吗?
RET
CLEAR ENDP
SCAN_KEY PROC NEAR
MOV DX,CMD_8279
IN AL,DX ;读状态
READ_FIFO: AND AL,7
JZ NO_KEY ;是否有键按下
READ: MOV AL,40H
OUT DX,AL ;读FIFO RAM
MOV DX,DATA_8279
IN AL,DX
STC ;有键
SCAN_KEY1: RET
NO_KEY: CLC ;无键按下,清CY
JMP SCAN_KEY1
SCAN_KEY ENDP

KEY_NUM PROC NEAR
AND AL,3FH
RET
KEY_NUM ENDP
```


WRITE_DATA PROC NEAR

MOV DX,DATA_8279

OUT DX,AL

RET

WRITE_DATA ENDP

;初始化 8259a

Init8259 PROC NEAR

MOV DX,I08259_0 ;ICW1 写入端口 0

MOV AL,13H ;10010B 脉冲触发,单片 8259a,不需要 icw4

OUT DX,AL

MOV DX,I08259_1 ;ICW2 写入端口 1

MOV AL,08H ;00001000B 中断类型号高五位 00001

OUT DX,AL

MOV AL,09H

OUT DX,AL

MOV AL,0FEH ;11111110B OCW1 2

OUT DX,AL

RET

Init8259 ENDP

WriIntver PROC NEAR

PUSH ES

MOV AX,0

MOV ES,AX

MOV DI,20H

LEA AX,INT_0

STOSW

MOV AX,CS

STOSW

POP ES

RET

WriIntver ENDP

;初始化 8255

INIT8255 PROC NEAR

MOV DX,COM_ADD55

MOV AL,80H

OUT DX,AL

MOV DX,PA_ADD55

MOV AL,0FFH

```

OUT DX,AL
RET
INIT8255 ENDP
;以下为点阵显示相关函数
INIT_IO      PROC NEAR
MOV DX,PC_ADD55    ;8255 控制字地址
MOV AL,80H    ;设置 8255 的 PA、PB、PC 口为输出口
OUT DX,AL    ;写控制字
MOV DX,ADDR_8155_C    ;8155 控制字地址
MOV AL,03H    ;设置 8155 的 PA 口为输出
OUT DX,AL    ;写控制字
RET
INIT_IO      ENDP
CLEARDISP PROC    NEAR
MOV AL,0FFH
MOV DX,LINE1
OUT DX,AL
MOV DX,LINE2
OUT    DX,AL
MOV AL,0
MOV DX,ROW1
OUT DX,AL
MOV DX,ROW2
OUT DX,AL
RET
CLEARDISP ENDP
ADJUST  PROC    NEAR
PUSH    CX
MOV CX,8
ADJUST1: RCL AL,1
XCHG AL,AH
RCR     AL,1
XCHG AL,AH
LOOP    ADJUST1
MOV     AL,AH
POP     CX
RET
ADJUST  ENDP
DISP_CH PROC NEAR

```

```
PUSH    CX
MOV     CX,8
DISP_CH_1:  CALL DISP1
LOOP    DISP_CH_1
POP     CX
RET
DISP_CH  ENDP
DISP1    PROC NEAR
PUSH    SI
PUSH    CX
MOV     CX,16    ;计数器,16 列依次被扫描
MOV     BL,0FEH    ;上边列输出值
MOV     BH,0FFH    ;下边列输出值
REPEAT:    MOV     DX,LINE1
MOV     AL,BL
OUT     DX,AL    ;上边列输出
MOV     DX,LINE2
MOV     AL,BH
OUT     DX,AL    ;下边列输出
LODSB
CALL    ADJUST    ;调整 AL,将 AL 中二进制数旋转 180 度
MOV     DX,ROW1
OUT     DX,AL    ;左边行输出
LODSB
CALL    ADJUST    ;调整 AL,将 AL 中二进制数旋转 180 度
MOV     DX,ROW2
OUT     DX,AL    ;右边行输出
CALL    DL10MS
CALL    CLEAR DISP
STC
RCL     BL,1
RCL     BH,1    ;循环移位 BX,行线扫描输出 0
LOOP    REPEAT
POP     CX
POP     SI
RET
DISP1    ENDP

INT_0:    PUSH DX
PUSH AX
```

```
XCHG AL,SWCOUNT
```

```
CMP AL,0
```

```
JZ SWITCH1
```

```
CMP AL,1
```

```
JZ SWITCH2
```

```
CMP AL,2
```

```
JZ SWITCH3
```

```
SWITCH1 PROC NEAR
```

```
LEA SI,PW1 ;切换为密码组 1
```

```
LEA DI,PW0
```

```
MOV CX,6
```

```
CLD
```

```
REPE MOVSB
```

```
INC AL
```

```
XCHG AL,SWCOUNT
```

```
MOV DX,I08259_0
```

```
MOV AL,20H
```

```
OUT DX,AL
```

```
POP AX
```

```
POP DX
```

```
IRET
```

```
SWITCH1 ENDP
```

```
SWITCH2 PROC NEAR
```

```
LEA SI,PW2 ;切换为密码组 2
```

```
LEA DI,PW0
```

```
MOV CX,6
```

```
CLD
```

```
REPE MOVSB
```

```
INC AL
```

```
XCHG AL,SWCOUNT
```

```
MOV DX,I08259_0
```

```
MOV AL,20H
```

```
OUT DX,AL
```

```
POP AX
```

POP DX

IRET

SWITCH2 ENDP

SWITCH3 PROC NEAR

LEA SI,PW3 ;切换为密码组 3

LEA DI,PW0

MOV CX,6

CLD

REPE MOVSB

MOV AL,0H

XCHG AL,SWCOUNT

MOV DX,I08259_0

MOV AL,20H

OUT DX,AL

POP AX

POP DX

IRET

SWITCH3 ENDP

MOV DX,I08259_0

MOV AL,20H

OUT DX,AL

POP AX

POP DX

IRET

;延时 500ms

DL500ms PROC NEAR

PUSH CX

MOV CX,55000

DL500ms1: LOOP DL500ms1

POP CX

RET

DL500ms ENDP

DL10ms PROC NEAR

PUSH CX

MOV CX,133

LOOP \$

```
POP    CX
RET
DL10ms  ENDP
;延时 5s
DL5S PROC NEAR
PUSH CX
MOV  CX,10
DL5S1:  CALL    DL500ms
LOOP DL5S1
POP  CX
RET
ENDP
;延时 1s
DL1S PROC NEAR
PUSH CX
MOV  CX,2
DL1S1:  CALL    DL500ms
LOOP DL5S1
POP  CX
RET
ENDP
END1:   END  START
```

3.4.2 控制器端

```
#include "common.h"
#include "include.h"

//函数申明
void output_pw1();
void output_pw2();
void output_pw3();
void output_timebar();
void count_down(byte t_count);

//变量申明
uint8 pw1[6] = {'1', '2', '3', '4', '5', '6'};
uint8 pw2[6] = {'1', '2', '3', '3', '2', '1'};
uint8 pw3[6] = {'6', '5', '4', '3', '2', '1'};
uint8 countdown[10] = {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9'};
uint8 pw_flag = 1;
uint8 timeset = 9;

//主函数
void main(void) {
    DisableInterrupts;
    initialization();
    EnableInterrupts; //数据采集完成
    LCD_P6x8Str(10, 0, "Password Reset in");
    while (1) {
        //显示第一个密码组并产生脉冲
        output_pw1();
        count_down(9);
        gpio_turn(PTD8);
        DELAY_MS(800);
        gpio_turn(PTD8);

        //显示第二个密码组并产生脉冲
        pw_flag++;
        output_pw2();
        count_down(9);
```

```
    gpio_turn(PTD8);
    DELAY_MS(800);
    gpio_turn(PTD8);
//显示第三个密码组并产生脉冲
    pw_flag++;
    output_pw3();
    count_down(9);
    gpio_turn(PTD8);
    DELAY_MS(800);
    gpio_turn(PTD8);
    pw_flag = 1;
}
} // main
//显示输出
void output_pw1() {
    for (int i = 0; i <= 5; i++) {
        LCD_write_char(10 * i + 35, 6, pw1[i]);
    }
}
void output_pw2() {
    {
        for (int i = 0; i <= 5; i++)
            LCD_write_char(10 * i + 35, 6, pw2[i]);
    }
}
void output_pw3() {
    for (int i = 0; i <= 5; i++) {
        LCD_write_char(10 * i + 35, 6, pw3[i]);
    }
}
//倒数计时
void count_down(uint8 t_count) {

    while (t_count != 0) {
        LCD_write_char(60, 2, countdown[t_count]);
        t_count--;
    }
}
```



```
    DELAY_MS(1000);  
}  
t_count = timeset;  
}
```

第四章 实验结果及分析

4.1 实验结果

测试项目	测试结果
待机	显示屏显示锁定标志
输入显示屏上的密码	显示屏显示解锁标志并播放解锁声音
当密码自动更换后输入新密码	显示屏显示解锁标志并播放解锁声音
随机输入错误密码	显示屏显示锁定标志并播放错误声音
随机输入三次错误密码	显示屏显示锁定标志并播放警告声音

4.2 分析

符合实验的预期要求，各功能模块均工作正常，基本实现设计要求。

在键盘输入密码时，LED 点阵会出现闪烁。但是由于时间限制，我并没有对此问题进行 debug。

第五章 课设体会和总结

课程设计是培养学生综合运用所学知识，确定课程设计的课题，然后展开一系列设计实验内容。接触《微机原理与接口技术》已经一个学期了，理论知识的学习基本已经完成。通过这次课程设计我对这门课有了一个新的认识。

这次课程设计对我最大的感触就是，汇编语言的编程难度远远超过了 C 语言等高级语言，使得实现功能的时间大大增加，开发时不仅要考虑算法及逻辑的问题，很大一部分精力要放在寄存器的配制上。脑子中很多很好的想法，由于试验平台的硬件和汇编语言的限制无法实现。这也体现出，高级编程语言对于推动计算机技术发展的重要性，使得程序员可以专注在算法和逻辑的实现上，同时也提高了程序的通用性。

由于我并没有挑选老师给定的题目，而是自己去规划我的项目，我就需要使用很多课上没有教授的芯片和模块。为了使用这些芯片和模块，我对实验系统说明书上提供的例程进行了大量的单步调试，观察寄存器和内存的变化情况。最后成功的移植到我的程序上。这项工作锻炼了我对于陌生模块快速上手的能力，这些经验可以运用在以后的开发工作中。

第六章 参考文献

- [1] 微型计算机原理与汇编语言程序设计 中国水利水电出版社 杨立主编 2003年2月第1版
- [2] 微机原理——结构，编程与接口 清华大学出版社 北京交通大学出版社 王克义编著 2009年3月第1版
- [3] 微机原理与接口技术 高等教育出版社 徐惠民 主编 2007年8月第1版
- [4] 微机原理与接口技术实验指导书 上海海事大学电工电子实验中心
- [5] 微机原理，汇编与接口技术学习指导 清华大学出版社 朱定华 编著 2006年12月第1版
- [6] 微型计算机原理及应用学习指导 化学工业出版社 赵邦信 林嵘 编著 2010年8月第1版
- [7] 微机原理接口技术实验月实践教程 清华大学出版社 汤书森 主编 2008年7月第1版
- [8] 微机原理及应用 高等教育出版社 陈继红 徐晨 王春明 徐慧 编著 2011年2月第2版
- [9] 微机接口技术与实例 国防工业出版社 李广波 王海 李岗 编著 2010年5月第1版