****

课程设计报告书

**题目：基于单片机的显示系统**

**学 院 电子与信息学院**

**专 业 信息工程3班**

**学生姓名 蒲 尧**

**学生学号 201630258438**

**指导教师 梁亚玲**

**课程编号**

**课程学分 1.0**

**起始日期 2018.5.12**

|  |  |
| --- | --- |
| 教  师  评  语 | 教师签名：  日期： |
| 成  绩  评  定 |  |
| 备  注 |  |

目录

[一、选题背景 1](#_Toc516837060)

[二、方案论证(设计理念) 1](#_Toc516837061)

[三、过程论述 3](#_Toc516837062)

[（一）分立模块 3](#_Toc516837063)

[（二）综合模块 19](#_Toc516837064)

[四、结果分析 29](#_Toc516837065)

[五、课程设计总结 36](#_Toc516837066)

**课程名称：基于单片机的显示系统**

## 一、选题背景

单片机课程设计是《微机系统与接口技术》课程学习中不可缺少的实践性教学环节。通过课程设计，加深学生对本门课程基本知识的理解，培养学生运用所学知识，分析和解决问题的技能，提高学生对课程基本知识的应用能力，训练并提高学生查阅文献、分析资料和撰写论文的基本功。单片微型计算机作为微型计算机的一个重要分支，在技术日趋成熟的今天，其灵活的硬件电路的设计和软件的设计，让它得到了广泛的应用，几乎是从小的电子产品，到大的工业控制，单片机都起到了举足轻重的作用。其中单片机用于电子产品的设计是非常方便的。

本实验要求如下：

1. 分立小模块功能要求
2. 输入按键并显示：4×4矩阵键盘分别定义为：0、1、2、3、4、5、……、E、F，16个按键。当键入一个按键，在一位数码管上显示键值。
3. 键盘控制LED：利用键盘0~7八个按键，分别控制8个LED。当键入其中一个按键，点亮相应的LED。
4. 外部中断0：按下KEY1，在四位数码管闪烁显示“ERR-”三次。
5. 外部中断1：按下KEY2，LED演示走马灯10次，并且LED点亮的时间控制必须精确定时，利用定时器定时时间。
6. 综合模块要求

将分立小模块的多个功能集成在一个项目中，自主设计一个具有实际应用功能的程序，鼓励自主创新。

## 二、方案论证(设计理念)

**1.键盘扫描**

（1）矩阵键盘的介绍

单片机中的矩阵键盘是由51芯片中的P2端口所拉出来的8根引线组成的，其中P2.0~P2.3作为行线，P2.4~P2.7作为列线。在行线和列线的每个交叉点上设置一个按键，这样键盘上按键的个数就为4×4个。

（2）识别矩形键盘按键的方法

对矩阵键盘按键识别常用的有扫描法，反极法和位操作法：

①扫描法

首先判断是否有键按下，使全部行线输出低电平，检测列线的电平状态；确认有键按下后，每条行线分别单独输出低电平，检测列线为零的交点，由此得到的一个八位二进制可利用穷举法进行键值赋值。

②反极法

行线全送出低电平，检测列线状态，出现低电平的列线即被按下；此后，相反的，将全部列线送出低电平，检测行线状态，出现低电平的行线即被按下。将得到的两个表示状态的八位二进制进行或运算，就在进行查找键值。

③位操作法

把行输出线逐位置为低电平，对列输入线逐位扫描，有键按下时列线变为低电平。再把行线逐条置为低电平，其余行线置为高电平，用位比较指令逐列扫描，确定按下的键所在的列。赋予键值。把行输出线逐位置为低电平，对列输入线用位比较指令逐位扫描，若列线全部为高电平，表示按键已释放。

三种方法比较而言，“反极法”最为简易快捷，代码短而又能够实现功能。故本实验采用反极法得到按键值，利用查表法进行按键赋值，为之后再次查表赋予数码管码值以及LED灯码值做好准备。

**2.74HC595**

74HC595是一个8位串行输入、并行输出的位移缓存器：并行输出为三态输出。在SCK (p0.5)的上升沿，串行数据由SDL输入到内部的8位位移缓存器，并由Q7'输出，而并行输出则是在LCK的上升沿将在8位位移缓存器的数据存入到8位并行输出缓存器。当串行数据输入端OE（p0.6）的控制信号为低使能时，并行输出端的输出值等于并行输出缓存器所存储的值。而当OE为高电位，也就是输出关闭时，并行输出端会维持在高阻抗状态。

所以通过控制p0.5和p0.6使按键值或者寄存器的值循环右移或者左移串行输入，并行输出到数码管。

**3.中断处理**

在AG嵌入式开发板V1.2中含有两个外部中断源，利用这两个中断源可以切换不同的功能，实验中通过按下中断0、中断1，可以分别执行不同的子功能。

按下中断0，定时器0开始计时，分别查表给4个数码管赋值‘E’‘R’‘R’‘-’通过定时器计时使单片机四位数码管闪烁显示“ERR-”，设置R1=5控制亮时间5\*0.2=1s，R1=3控制暗时间3\*0.2=0.6s。

按下中断1，定时器0计时，设置R7=4使亮灯间隔 4\*（0FFFFH-3CB0）μs≈0.2s，总用时80\*0.2≈16s。

## 三、过程论述

#### （一）分立模块

**1.****数码管显示键盘输入数值**

运用反极法扫描键盘，先高四位清0，将高四位通过累加器A存到R0；再低四位清0，低四位通过累加器A与R0或将键值存到R0，通过查键盘表将其转化为0~15存至R5再赋给A用来查数码管表，通过控制74HC595的p0.4,p0.5,p0.6来串行输入并行输出至数码管，通过给p0.0~p0.3赋值控制所亮的数码管是哪一个。

**流程图：**

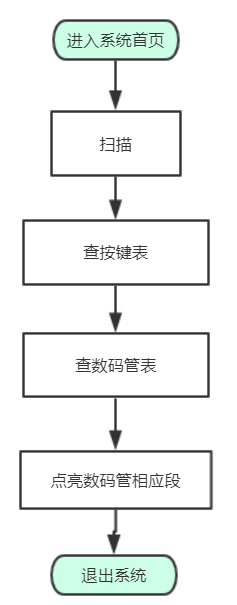
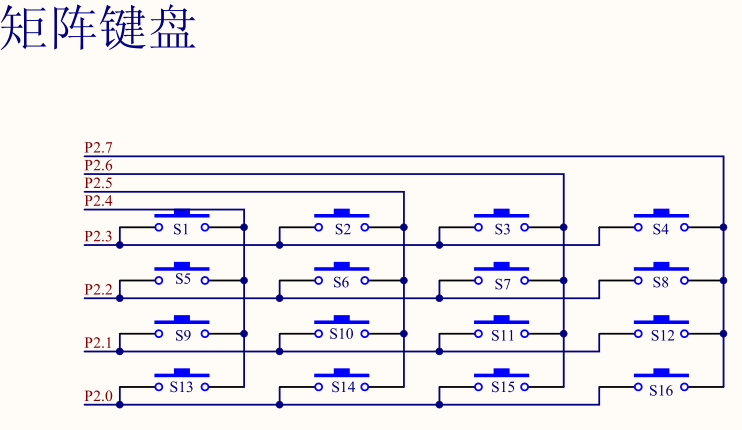


图3.1数码管显示键盘输入数值流程图

**用到的原件图**



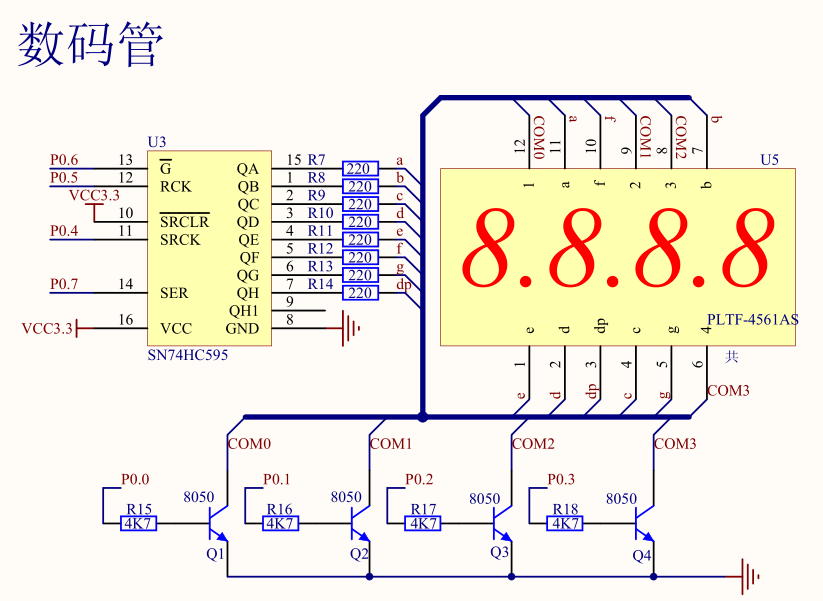
图3.2矩阵键盘

图3.3 AG嵌入式开发板V1.2数码管和74HC595

**功能1单独程序源代码：**

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*按键显示\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ORG 0000H

AJMP MAIN

ORG 000BH

MAIN**:** **NOP**

;———————————————反极法扫描键盘——————————————————

;————— ——————————按键列查询—————————————————————————

KEYC**:** **MOV** P2**,**#0FH ;按键高四位即行清0

**MOV** A**,**P2

ANL A**,**#0FH ;累加器A高四位即行清0

CJNE A**,**#0FH**,**KC1 ;判断是否按下按键，是则跳至KC1存值，否则再次扫描

AJMP KEYC

KC1**:** LCALL DLY ;调用延时子程序

**MOV** A**,**P2 ;将键值存入A

ANL A**,**#0FH ;累加器A高四位即行再次清0

CJNE A**,**#0FH**,**KC2 ;判断是否按下按键，是则跳至KC2存值，否则再次扫描

AJMP KEYC

KC2**:** **MOV** R0**,**A ;将列键值A存入R0

;——————————————按键行查询—————————————————————————

KEYR**:** **MOV** P2**,**#0F0H ;按键低四位即列清0

**MOV** A**,**P2

ANL A**,**#0F0H ;累加器A低四位即列清0

CJNE A**,**#0F0H**,**KR1

AJMP KEYR

KR1**:** LCALL DLY

**MOV** A**,**P2

ANL A**,**#0F0H

CJNE A**,**#0F0H**,**KR2

AJMP KEYR

KR2**:** ORL A**,**R0

**MOV** R0**,**A ;将行键值A与原R0或，存入R0包含行与列

AJMP CHECK

;————————————————延时子程序1————————————————————————

;1+30+2\*150\*30+2\*30+2=9151US

DLY**:** **MOV** R7**,**#30

DLY1**:** **MOV** R6**,**#150

DLY2**:** DJNZ R6**,**DLY2

DJNZ R7**,**DLY1

**RET**

;————————————————将键值转换成数码管码值————————————————

CHECK**:** **MOV** A**,**#00H

**MOV** R5**,**#00H

NEXT1**:** **MOV** A**,**R5

**INC** A ;A不断增加直到与R0相等才不再循环，最终A为键值，R5为按键数值

**MOV** R5**,**A

**MOV** DPTR**,**#TABLE1 ;装入键值表头

MOVC A**,**@A**+**DPTR ;按键值与键值表比较

CLR CY ;进位标志CY清0

SUBB A**,**R0 ;(A)<-(A)-(R0)-(CY)

**JNZ** NEXT1 ;A不等于0则循环查表

NOW**:** **MOV** P0**,**#01H ;

**MOV** R3**,**#0FH ;R3高位置0，控制串行输入循环

**MOV** DPTR**,**#TABLE2 ;装入数码管表头

**MOV** A**,**R5 ;将按键数值存入A

MOVC A**,**@A**+**DPTR ;按键值与数码管表比较并将数码管值存入A

;———————————————延时2——————————————————

;1+2\*20=41US

DLY3**:** **MOV** R6**,**#20H

DLY4**:** DJNZ R6**,**DLY4

;————————————————SN74HC595操作———————————————————

**LOCK:** CPL P0.4 ;P0.4位位取反

**JB** P0.4**,**NEXT2 ;P0.4=1,则跳至NEXT2

RL A ;A左移，即A\*2

**MOV** **C,**ACC.7 ;把A的第8位给74HC595的串行输入端P0.7

**MOV** P0.7**,C**

NEXT2**:** DJNZ R3**,**DLY3 ;R3自减不为0则循环向P0.7输出

**SETB** P0.5 ;P0.5置1

CLR P0.5 ;P0.5清0

AJMP KEYC ;循环按键列查询

;————————————————键值表与数码管表—————————————————————

TABLE1**:** DB 0E7H**,**0EBH**,**0EDH**,**0EEH**,**0D7H**,**0DBH**,**0DDH**,**0DEH**,**0B7H**,**0BBH**,**0BDH**,**0BEH**,**77H**,**7BH**,**7DH**,**7EH

TABLE2**:** DB 3FH**,**06H**,**5BH**,**4FH**,**66H**,**6DH**,**7DH**,**07H**,**7FH**,**6FH**,**077H**,**07CH**,**039H**,**05EH**,**079H**,**071H

;数码管：0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

END

**2.****按键点亮相应数字的LED灯**

反极法扫描键盘，查表输给p0点亮相应的灯，这个问题思路比较简单，就是循环查赋值。

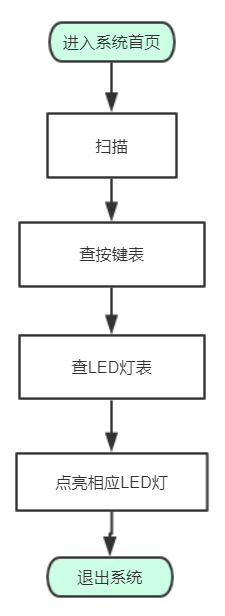
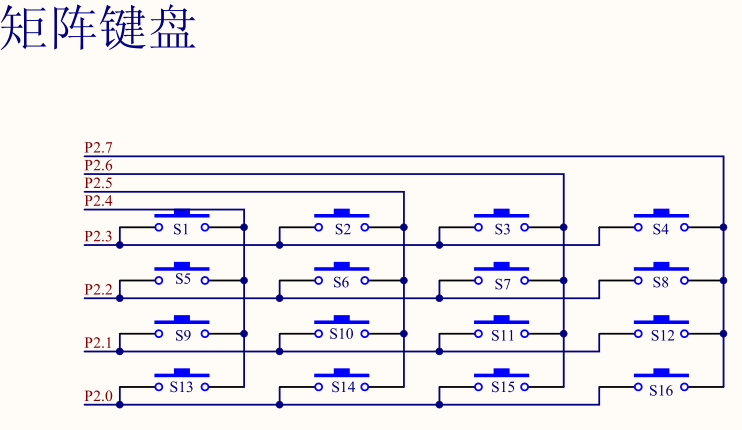
**流程图**

图3.4按键点亮相应数字的LED灯流程图

**用到的原件图**

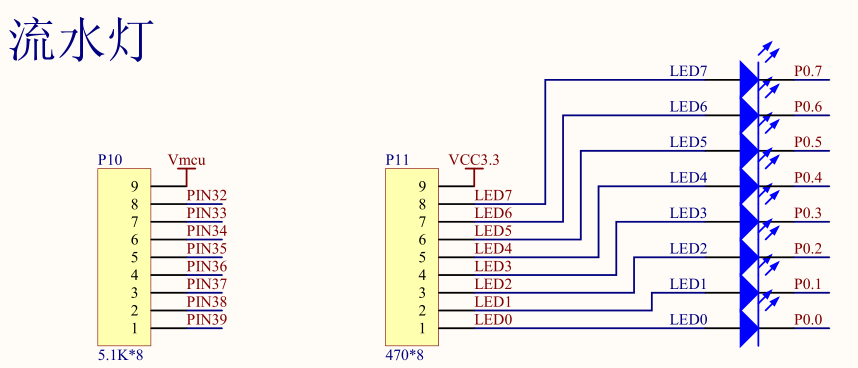
图3.5矩阵键盘

图3.6流水灯

**功能2单独程序源代码**

ORG 0000H

AJMP MAIN

ORG 000BH

MAIN**:** **NOP**

;———————————————————按键列查询——————————————————

KEYC**:** **MOV** P2**,**#0FH ;按键高四位即行清0

**MOV** A**,**P2

ANL A**,**#0FH ;累加器A高四位即行清0

CJNE A**,**#0FH**,**KC1 ;判断是否按下按键，是则跳至KC1存值，否则再次扫描

AJMP KEYC

KC1**:** LCALL DLY ;调用延时子程序

**MOV** A**,**P2 ;将键值存入A

ANL A**,**#0FH ;累加器A高四位即行再次清0

CJNE A**,**#0FH**,**KC2 ;判断是否按下按键，是则跳至KC2存值，否则再次扫描

AJMP KEYC

KC2**:** **MOV** R0**,**A ;将列键值A存入R0

;————————————————按键行查询———————————————————

KEYR**:** **MOV** P2**,**#0F0H ;按键低四位即列清0

**MOV** A**,**P2

ANL A**,**#0F0H ;累加器A低四位即列清0

CJNE A**,**#0F0H**,**KR1

AJMP KEYR

KR1**:** LCALL DLY

**MOV** A**,**P2

ANL A**,**#0F0H

CJNE A**,**#0F0H**,**KR2

AJMP KEYR

KR2**:** ORL A**,**R0

**MOV** R0**,**A ;将行键值A与原R0或，存入R0包含行与列

AJMP CHECK

;——————————————————延时子程序——————————————————

;1+30+2\*150\*30+2\*30+2=9151US

DLY**:** **MOV** R7**,**#30

DLY1**:** **MOV** R6**,**#150

DLY2**:** DJNZ R6**,**DLY2

DJNZ R7**,**DLY1

**RET**

;—————————————————将键值传至LED——————————————

CHECK**:** **MOV** A**,**#00H

**MOV** R5**,**#00H

NEXT1**:** **MOV** A**,**R5

**INC** A

**MOV** R5**,**A ;将符合的键值暂存入R5

**MOV** DPTR**,**#TABLE1 ;装入键值表头

MOVC A**,**@A**+**DPTR ;按键值与键值表比较

CLR CY ;进位标志CY清0

SUBB A**,**R0 ;(A)<-(A)-(R0)-(CY)

**JNZ** NEXT1 ;A不等于0则循环查表

NOW**:** **MOV** DPTR**,**#TABLE3 ;装入LED表头

**MOV** A**,**R5 ;将键值存入A

MOVC A**,**@A**+**DPTR ;按键值与数码管表比较

**MOV** P0**,**A ;符合的键值存入P0点亮LED灯

AJMP KEYC ;循环按键列查询

;—————————————————键值表,LED表—————————————————

;TABLE1: DB 00H

; DB 0E7H,0D7H,0B7H,77H,0EBH,0DBH,0BBH,7BH,0EDH,0DDH,0BDH,7DH,0EEH,0DEH,0BEH,7EH

; 4X4矩阵键盘：0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F(左下到右上）

TABLE1**:** DB 0E7H**,**0EBH**,**0EDH**,**0EEH**,**0D7H**,**0DBH**,**0DDH**,**0DEH**,**0B7H**,**0BBH**,**0BDH**,**0BEH**,**77H**,**7BH**,**7DH**,**7EH

TABLE3**:** DB 07FH**,**0BFH**,**0DFH**,**0EFH**,**0F7H**,**0FBH**,**0FDH**,**0FEH**,**07FH**,**0BFH**,**0DFH**,**0EFH**,**0F7H**,**0FBH**,**0FDH**,**0FEH ;对应LED键字0,1,2,3,4,5,6,7,0,1,2,3,4,5,6,7

END

**3.中断0显示“ERR-”**

这个部分是队友做的，但我也有了解，主要思路是点亮数码管时用了延时，灭数码管时也用了延时。利用定时器0延时点亮时间5\*（4\*0.05）=1s，暗淡时间3\*（4\*0.05）=0.6s。

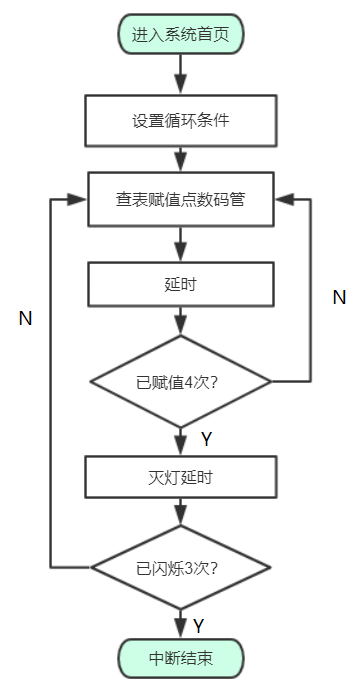
**流程图**

图3.7中断0流程图

**中断用到的原件**

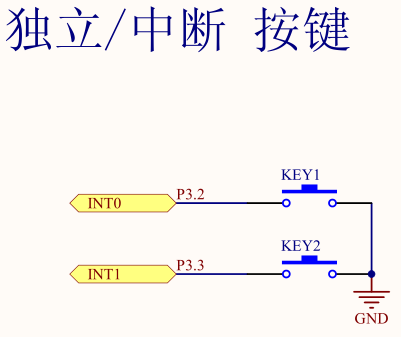


图3.8独立中断按键

**功能3单独程序源代码**

ORG 0000H

AJMP MAIN

ORG 000BH ;定时器溢出

AJMP TIMER

ORG 0003H ;外部中断0

AJMP **ERROR**

ORG 0030H

MAIN**:**

**SETB** EA ;全局中断

**SETB** EX0 ;外部中断0中断

**SETB** EX1 ;外部中断1中断

**SETB** PT0 ;定时器0：高优先级

CLR PX0 ;外中断0：低优先级

CLR PX1 ;外中断1：低优先级

**NOP**

;=============================================== 数码管显示

TUBE**:**

**MOV** R6**,**#00H

**MOV** P0**,**R6

**MOV** R4**,**#08H

CLR P0.6

PUT\_IN**:**

RRC A ;带进位累加器循环右移

**MOV** P0.7**,C**

**SETB** P0.4

CLR P0.4

DJNZ R4**,**PUT\_IN

AJMP PUT\_OUT

PUT\_OUT**:**

**SETB** P0.5

CLR P0.5

**RET**

;============================================= 延时消除抖动

DELAYS**:**

**MOV** R6**,**#50

D1S1**:**

**MOV** R5**,**#50

DJNZ R5**,$**

DJNZ R6**,**D1S1

**RET**

;================================================== 短延时

DELAYX**:**

**MOV** R6**,**#0FFH

D1X1**:**

**MOV** R5**,**#0FFH

DJNZ R5**,$**

DJNZ R6**,**D1X1

**RET**

;============================================== 延时0.1s

DELAY**:**

**MOV** TMOD**,**#01H ;选择工作模式1，定时器0 自动重装初值的8位定时器

**MOV** TH0**,**#3CH ;初值高字节

**MOV** TL0**,**#0B0H ;初值低字节

**SETB** ET0 ;定时器T0中断允许

**SETB** TR0 ;启动定时器T0

CLR F0

**MOV** R7**,**#02H

**RET**

;=============================================== 定时器中断

TIMER**:**

**MOV** TMOD**,**#01H

**MOV** TH0**,**#3CH

**MOV** TL0**,**#0B0H

DJNZ R7**,**RTI

**MOV** R7**,**#02H

**SETB** F0

RTI**:**

RETI

;====================================外部中断0，数码管闪烁Err-

**ERROR:**

LCALL DELAY

**MOV** R0**,**#03H

**MOV** R1**,**#05H

SHOW**:**

**MOV** R5**,**#9EH

**MOV** A**,**R5

LCALL TUBE

**SETB** P0.0

LCALL DELAYS

CLR P0.0

**MOV** R5**,**#0AH

**MOV** A**,**R5

LCALL TUBE

**SETB** P0.1

LCALL DELAYS

CLR P0.1

**MOV** R5**,**#0AH

**MOV** A**,**R5

LCALL TUBE

**SETB** P0.2

LCALL DELAYS

CLR P0.2

**MOV** R5**,**#02H

**MOV** A**,**R5

LCALL TUBE

**SETB** P0.3

LCALL DELAYS

CLR P0.3

**JNB** F0**,**SHOW

CLR F0

DJNZ R1**,**SHOW

**MOV** R1**,**#03H

**MOV** P0**,**#00H

BLANK**:**

**JNB** F0**,**BLANK

CLR F0

DJNZ R1**,**BLANK

**MOV** R1**,**#05H

DJNZ R0**,**SHOW

RETI

;==================================================

END

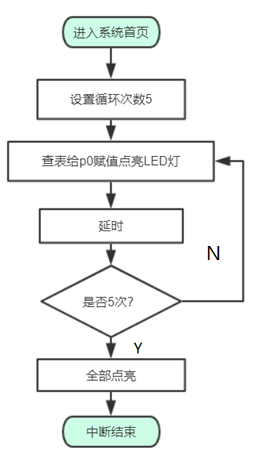
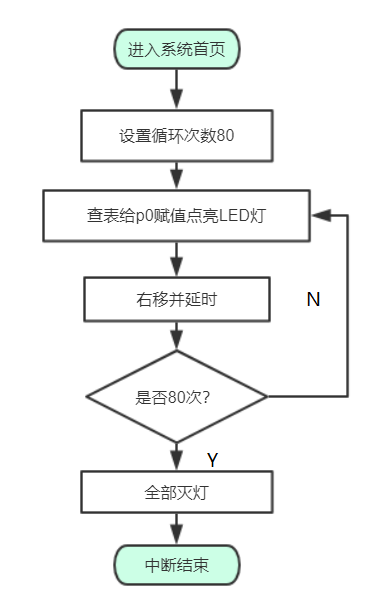
**4.中断1控制10次走马灯**

单独模块：不断查表，左右来回走马灯，左右算一次的话，附循环次数为5即可。

综合模块：走马灯比较简单，但是用到定时器就比较复杂一点了。走马灯思路是设置初始p0为7FH，然后不断右移即可；定时器思路是设置TH0和TL0，使一次中端用时约0.05s，我给R7置4为循环次数，所以综合模块延时程序为延时0.05\*4=0.2s，走马灯10次，也就是把p0的值不断右移80次，用时约80\*0.2=16s。实验结果符合预期。

**流程图**

2种方法

图3.9两种LED灯走马灯流程图

**功能4单独程序源代码**

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*1中断走马灯\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ExtInt1 EQU P3.3

ORG 0000H

**JMP** MAIN ;主程序入口

ORG 0013H

AJMP INT1\_SER ;外部中断1服务子程序入口

ORG 001BH

AJMP TIME1\_SER ;计时器T1服务子程序入口

ORG 0030H

MAIN**:** **MOV** **SP,**#60H ;设置堆栈长度为60byte

**MOV** P0**,**#8FH

**SETB** ET1 ;允许计时器T1溢出中断

**SETB** ExtInt1 ;将p3.3即外部中断1置1

L10**:** LCALL INT1\_INIT ;调用外部中断1初始化子程序

LCALL TIME1\_INIT ;调用计时器1初始化子程序

AJMP L10

;———————————————外部中断1初始化子程序-(课本p72）—————————————

INT1\_INIT**:**

**SETB** IT1 ;INT1中断触发方式为边沿触发

**SETB** EX1 ;外部中断1允许

**SETB** EA ;全局中断允许

CLR TF1 ;溢出标志位清0

CLR TR1 ;T1清0，停止计时器T1

**RET**

;—————————————定时器1初始化子程序-（课本P63）——————————————

TIME1\_INIT**:**

**MOV** TMOD**,**#90H ;采用方式1，T1受INT1控制

**MOV** TH1**,**#00H ;预置初值高低位全为0

**MOV** TL1**,**#00H

**SETB** ET1 ;定时器1中断允许

**SETB** PT1 ;定时中断1优先置1

**RET**

;————————————————中断1服务函数—————————————————

INT1\_SER**:**

**MOV** R1**,**#00

**MOV** R2**,**#02

**MOV** R3**,**#05 ;走马灯来回次数，一个来回记一次

START**:** **SETB** TR1 ;定时器1开始计数

NOT\_FINISH**:**

**JB** TF1**,**NOT\_FINISH;TF=1,则一直循环

**JNB** ET1**,**K40 ;ET1=0,定时器1中断不允许，则结束

AJMP START ;定时器1中断允许，则再次计数

K40**:**

CLR EX1 ;外部中断1清0

**MOV** P0**,**#00

RETI

;———————————————定时器1服务函数————————————————

TIME1\_SER**:**

DJNZ R2**,**RETURN ;R1递增,R3递减一起控制循环次数

**MOV** A**,**R1

**MOV** DPTR**,**#TABLE ;表头

MOVC A**,**@A**+**DPTR ;按键值与led键值比较

**MOV** P0**,**A ;符合的键值存入P0点亮相应的LED灯

**MOV** R2**,**#02 ;每两次循环溢出后才点亮后一盏LED灯，每个灯点亮时间60ms\*2=0.12s

**INC** R1

CJNE R1**,**#14**,**RETURN ;R1不为14则退出子程序

**MOV** R1**,**#00

DJNZ R3**,**RETURN

CLR ET1 ;定时器1中断不允许

RETURN**:** RETI ;从中断程序中返回

TABLE**:** DB 07FH**,**0BFH**,**0DFH**,**0EFH**,**0F7H**,**0FBH**,**0FDH**,**0FEH**,**0FDH**,**0FBH**,**0F7H**,**0EFH**,**0DFH**,**0BFH

;对应led键字0,1,2,3,4,5,6,7,6,5,4,3,2,1

END

**5.按键控制蜂鸣器播放生日歌**

这里为了不与定时器0冲突，采用定时器1计时，不断查歌曲乐谱表，然后根据X=65536-（ftime/2/fmusic）得到的预置值表，赋予定时器1初值定时结束反转p1.3电平即可控制音调

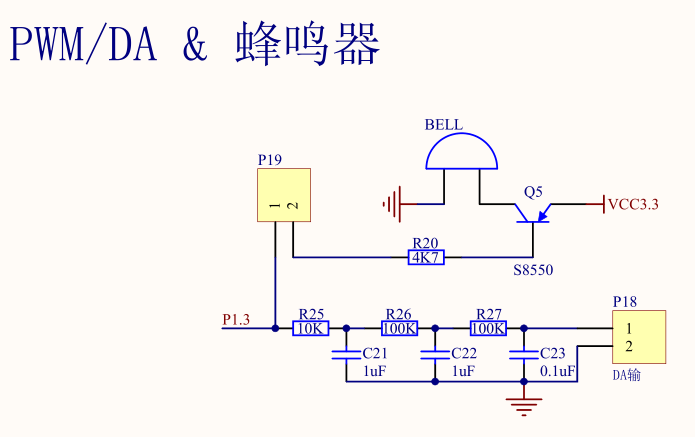
**原件**

图3.10蜂鸣器

**功能5单独程序源代码**

ORG 0000H

**JMP** START

ORG 001BH

**JMP** TIME1

ORG 0030H

START**:**

**MOV** TMOD**,**#10H

**MOV** IE**,**#88H

START0**:**

**MOV** R1**,**#00H

NEXT**:**

**MOV** A**,**R1

**MOV** DPTR**,**#TABLE5

MOVC A**,**@A**+**DPTR

**MOV** R2**,**A

**JZ** END0

ANL A**,**#0FH

**MOV** R5**,**A

**MOV** A**,**R2

SWAP A ;高四位和低四位互换

ANL A**,**#0FH

**JNZ** SING

CLR TR1

**JMP** D1

SING**:**

**DEC** A

**MOV** R6**,**A

RL A

**MOV** DPTR**,**#TABLE6

MOVC A**,**@A**+**DPTR ;查表预置高位

**MOV** TH1**,**A

**MOV** 51H**,**A

**MOV** A**,**R6

RL A ;设定字节间隔

**INC** A

MOVC A**,**@A**+**DPTR ;查表预置低位

**MOV** TL1**,**A

**MOV** 50H**,**A

**SETB** TR1 ;启动定时器1

D1**:**

LCALL DELAYMUSIC

**INC** R1

**JMP** NEXT

END0**:**

CLR TR1

**JMP** START0

TIME1**:**

**PUSH** ACC ;保护ACC值

**PUSH** PSW ;保护PSW值

**MOV** TH1**,**51H

**MOV** TL1**,**50H

CPL P1.3 ;波形输出反相

**POP** PSW

**POP** ACC

RETI

DELAYMUSIC**:**

**MOV** R7**,**#02

D2**:**

**MOV** R4**,**#187

D3**:**

**MOV** R3**,**#248

DJNZ R3**,$**

DJNZ R4**,**D3

DJNZ R7**,**D2

DJNZ R5**,**DELAYMUSIC

**RET**

;歌曲表单

TABLE5**:**

DB 82H**,**01H**,**81H**,**94H**,**84H**,**0B4H**,**0A4H**,**04H

DB 82H**,**01H**,**81H**,**94H**,**84H**,**0C4H**,**0B4H**,**04H

DB 82H**,**01H**,**81H**,**0F4H**,**0D4H**,**0B4H**,**0A4H**,**94H

DB 0E2H**,**01H**,**0E1H**,**0D4H**,**0B4H**,**0C4H**,**0B4H**,**04H

DB 82H**,**01H**,**81H**,**94H**,**84H**,**0B4H**,**0A4H**,**04H

DB 82H**,**01H**,**81H**,**94H**,**84H**,**0C4H**,**0B4H**,**04H

DB 82H**,**01H**,**81H**,**0F4H**,**0D4H**,**0B4H**,**0A4H**,**94H

DB 0E2H**,**01H**,**0E1H**,**0D4H**,**0B4H**,**0C4H**,**0B4H**,**04H

DB 00H

;生日歌简谱：5 5 6 5 1 7 -| 5 5 6 5 2 1 -| 5 5 5 3 1 7 6 | 4 4 3 1 2 1 -

;音调

TABLE6**:**

DW 64260**,**64400**,**64524**,**64580

DW 64684**,**64777**,**64820**,**64898

DW 64968**,**65030**,**65058**,**65110

DW 65157**,**65178**,**65217

;低5，低6，低7，中1

;中2，中3，中4，中5

;中6，中7，高1，高2

;高3，高4，高5

END

#### （二）综合模块

功能1,2,4以及综合是我做的，所以对其中问题及解决措施还是比较了解的。

综合模块，首先，开头就要设置好，各个中断定时器的入口地址。

然后由于数码管和LED灯公用了p0，所以我用键盘扫描的方法选择功能，左下角第一个，则选择点亮数码管，接着就可以按键显示0~F了。reset重置后，倘若按到了左下角第二个，则选择点亮LED灯功能。中断0显示err-部分经过我的修改，我采用了循环查表的方式赋值。中断1则中规中矩，调用定时器0定时流水灯跳到下一个的间隔，再循环右移80次即可。

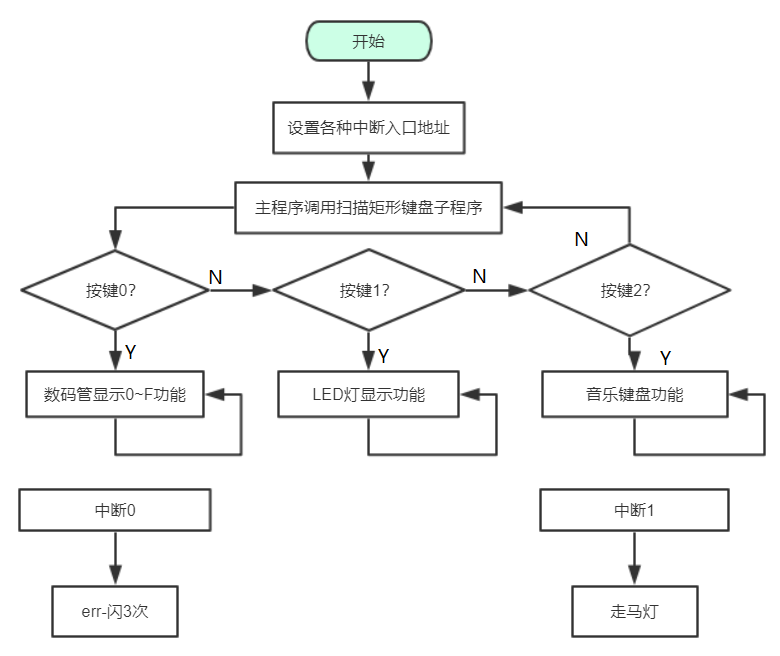
**流程图**

图3.11综合模块流程图

**综合程序源代码**

ORG 0000H ;主程序入口

AJMP MAIN

ORG 0003H ;外部中断0

AJMP ERR\_FLASH

ORG 000BH ;定时器0

AJMP TIMER0

ORG 0013H ;外部中断1

AJMP FLOW\_LIGHT

ORG 001BH ;定时器1

AJMP TIME1

ORG 0030H

MAIN**:**

**SETB** EA ;全局中断允许

**SETB** EX0 ;外部中断0中断允许

**SETB** EX1 ;外部中断1中断允许

**SETB** PT0 ;定时器0：高优先级

CLR PT1 ;定时器1：低优先级

CLR PX0 ;外中断0：低优先级

CLR PX1 ;外中断1：低优先级

;——————————开机闪烁作者1次——————————

LCALL DLY4

**MOV** R1**,**#05H ;延时

SHOW0**:**

**MOV** R5**,**#0CEH ;赋值P

**MOV** A**,**R5

LCALL TUBE

**SETB** P0.0

LCALL DLY3

CLR P0.0

**MOV** R5**,**#76H ;赋值Y

**MOV** A**,**R5

LCALL TUBE

**SETB** P0.1

LCALL DLY3

CLR P0.1

**MOV** R5**,**#0EEH ;赋值A

**MOV** A**,**R5

LCALL TUBE

**SETB** P0.2

LCALL DLY3

CLR P0.2

**MOV** R5**,**#0FCH ;赋值O

**MOV** A**,**R5

LCALL TUBE

**SETB** P0.3

LCALL DLY3

CLR P0.3

**JNB** F0**,**SHOW0

CLR F0

DJNZ R1**,**SHOW0

**MOV** R1**,**#03H

**MOV** P0**,**#00H ;灭数码管灯

;——————————根据键值判断选择功能——————————

LCALL SCAN ;扫描键盘

CJNE R0**,**#0E7H**,**COM1

AJMP CHECK1 ;按键为左下角第一个则选择CHECK1数码管显示数字功能

COM1**:**

CJNE R0**,**#0EBH**,**COM2;

AJMP CHECK2 ;按键为左下角从左到右第二个则选择CHECK2按键控制相应LED灯亮功能

COM2**:**

CJNE R0**,**#0EDH**,**MAIN;不为这3个按键则返回主程序重新扫描

AJMP ALARM

;===============功能1：数码管显示数值===============

CHECK1**:**

LCALL SCAN

;LCALL ALARM

**MOV** A**,**#00H

**MOV** R1**,**#00H

NEXT1**:**

**MOV** A**,**R1

**INC** A ;A不断增加直到与R0相等才不再循环，最终A为键值，R1为按键数值

**MOV** R1**,**A

**MOV** DPTR**,**#TABLE1 ;装入键值表头

MOVC A**,**@A**+**DPTR ;按键值与键值表比较

CLR CY ;进位标志CY清0

SUBB A**,**R0 ;(A)<-(A)-(R0)-(CY)

**JNZ** NEXT1 ;A不等于0则循环查表

NOW**:**

**MOV** R3**,**#08H ;R3高位置0，控制串行输入循环

**MOV** DPTR**,**#TABLE2 ;装入数码管表头

**MOV** A**,**R1 ;将按键数值存入A

MOVC A**,**@A**+**DPTR ;按键值与数码管表比较并将数码管值存入A

LCALL TUBE ;调用传输数据给数码管功能子程序

**SETB** P0.0 ;点亮第一个数码管

**SETB** P0.3

LCALL DLY3

;CLR P0.0 ;加上这句则为闪亮一下

AJMP CHECK1 ;返回CHECK1循环扫描

;===============功能2：按键控制LED灯显示===============

CHECK2**:**

LCALL SCAN

**MOV** A**,**#00H

**MOV** R2**,**#00H

NEXT3**:**

**MOV** A**,**R2

**INC** A

**MOV** R2**,**A ;将符合的键值暂存入R2

**MOV** DPTR**,**#TABLE1 ;装入键值表头

MOVC A**,**@A**+**DPTR ;按键值与键值表比较

CLR CY ;进位标志CY清0

SUBB A**,**R0 ;(A)<-(A)-(R0)-(CY)

**JNZ** NEXT3 ;A不等于0则循环查表

NOW1**:**

**MOV** DPTR**,**#TABLE3 ;装入LED表头

**MOV** A**,**R2 ;将键值存入A

MOVC A**,**@A**+**DPTR ;按键值与数码管表比较

**MOV** P0**,**A ;符合的键值存入P0点亮LED灯

AJMP CHECK2 ;返回CHECK2循环扫描

;===============功能3：外部中断0，数码管ERR-闪烁3次===============

ERR\_FLASH**:**

LCALL DLY4

**MOV** R0**,**#03H ;闪烁次数

**MOV** R1**,**#05H

**MOV** R2**,**#4

**MOV** R3**,**#0

**MOV** DPTR**,**#TABLE4

SHOW**:**

**MOV** A**,**R3

MOVC A**,**@A**+**DPTR ;查表

LCALL TUBE

SHU0**:** CJNE R3**,**#0**,**SHU1 ;判断是第1个字则点亮第1个数码管

**SETB** P0.0

LCALL DLY3

CLR P0.0

AJMP CMPR2

SHU1**:** CJNE R3**,**#1**,**SHU2 ;判断是第2个字则点亮第2个数码管

**SETB** P0.1

LCALL DLY3

CLR P0.1

AJMP CMPR2

SHU2**:** CJNE R3**,**#2**,**SHU3 ;判断是第3个字则点亮第3个数码管

**SETB** P0.2

LCALL DLY3

CLR P0.2

AJMP CMPR2

SHU3**:** CJNE R3**,**#3**,**CMPR2 ;判断是第4个字则点亮第4个数码管

**SETB** P0.3

LCALL DLY3

CLR P0.3

CMPR2**:**

**INC** R3

DJNZ R2**,**SHOW

**MOV** R2**,**#4

**MOV** R3**,**#0

**JNB** F0**,**SHOW

CLR F0

**MOV** R2**,**#4

**MOV** R3**,**#0

DJNZ R1**,**SHOW

**MOV** R1**,**#03H

**MOV** P0**,**#00H ;灭数码管灯

BLANK**:**

**JNB** F0**,**BLANK

CLR F0

DJNZ R1**,**BLANK

**MOV** R1**,**#05H

**MOV** R2**,**#4

**MOV** R3**,**#0

DJNZ R0**,**SHOW

CLR TR0

RETI

;===============功能4：外部中断1，LED演示走马灯(用到了定时器，总共用时约16s）===============

FLOW\_LIGHT**:**

LCALL DLY4 ;调用定时器计时子程序

**MOV** R3**,**#80 ;共右移80次

**MOV** A**,**#7FH ;设置初始值左边第一个灯亮

RUN**:**

**MOV** P0**,**A

**JNB** F0**,**RUN ;计时未完成则依旧点亮原来的灯

CLR F0

RR A ;循环右移

DJNZ R3**,**RUN

CLR TR0

**MOV** P0**,**#0FFH

RETI

;===============功能5（附加功能）：循环播放生日歌===============

ALARM**:**

**MOV** TMOD**,**#10H

**MOV** IE**,**#88H

START0**:**

**MOV** R1**,**#00H

NEXT**:**

**MOV** A**,**R1

**MOV** DPTR**,**#TABLE5

MOVC A**,**@A**+**DPTR

**MOV** R2**,**A

**JZ** END0

ANL A**,**#0FH

**MOV** R5**,**A

**MOV** A**,**R2

SWAP A ;高四位和低四位互换

ANL A**,**#0FH

**JNZ** SING

CLR TR1

**JMP** D1

SING**:**

**DEC** A

**MOV** R6**,**A

RL A

**MOV** DPTR**,**#TABLE6

MOVC A**,**@A**+**DPTR ;查表预置高位

**MOV** TH1**,**A

**MOV** 51H**,**A

**MOV** A**,**R6

RL A ;设定字节间隔

**INC** A

MOVC A**,**@A**+**DPTR ;查表预置低位

**MOV** TL1**,**A

**MOV** 50H**,**A

**SETB** TR1 ;启动定时器1

D1**:**

LCALL DELAYMUSIC

**INC** R1

**JMP** NEXT

END0**:**

CLR TR1

**JMP** START0

TIME1**:**

**PUSH** ACC ;保护ACC值

**PUSH** PSW ;保护PSW值

**MOV** TH1**,**51H

**MOV** TL1**,**50H

CPL P1.3 ;波形输出反相

**POP** PSW

**POP** ACC

RETI

DELAYMUSIC**:**

**MOV** R7**,**#02

D2**:**

**MOV** R4**,**#187

D3**:**

**MOV** R3**,**#248

DJNZ R3**,$**

DJNZ R4**,**D3

DJNZ R7**,**D2

DJNZ R5**,**DELAYMUSIC

**RET**

;===============功能5结束===============

;===============反极法扫描键盘函数===============

SCAN**:**

**NOP**

;——————————列查询——————————

KEYC**:**

**MOV** P2**,**#0FH ;按键高四位即行清0

**MOV** A**,**P2

ANL A**,**#0FH ;累加器A高四位即行清0

CJNE A**,**#0FH**,**KC1 ;判断是否按下按键，是则跳至KC1存值，否则再次扫描

AJMP KEYC

KC1**:**

LCALL DLY3 ;调用延时子程序

**MOV** A**,**P2 ;将键值存入A

ANL A**,**#0FH ;累加器A高四位即行再次清0

CJNE A**,**#0FH**,**KC2 ;再次判断是否按下按键，是则跳至KC2存值，否则再次扫描

AJMP KEYC

KC2**:**

**MOV** R0**,**A ;将列键值A存入R0

;——————————行查询——————————

KEYR**:**

**MOV** P2**,**#0F0H ;按键低四位即列清0

**MOV** A**,**P2

ANL A**,**#0F0H ;累加器A低四位即列清0

CJNE A**,**#0F0H**,**KR1

AJMP KEYR

KR1**:**

LCALL DLY3

**MOV** A**,**P2

ANL A**,**#0F0H

CJNE A**,**#0F0H**,**KR2

AJMP KEYR

KR2**:**

ORL A**,**R0

**MOV** R0**,**A ;将行键值A与原R0或，存入R0包含行与列

**RET**

;===============扫描键盘结束===============

;——————————74HC595传数据给数码管——————————

TUBE**:**

**MOV** P0**,**#00H

**MOV** R4**,**#08H

CLR P0.6

PUT\_IN**:**

RRC A ;带进位累加器循环右移

**MOV** P0.7**,C**

**SETB** P0.4

CLR P0.4

DJNZ R4**,**PUT\_IN

AJMP PUT\_OUT

PUT\_OUT**:**

**SETB** P0.5

CLR P0.5

**RET**

;——————————定时器0中断——————————

TIMER0**:**

**MOV** TMOD**,**#01H

**MOV** TH0**,**#3CH

**MOV** TL0**,**#0B0H

DJNZ R7**,**TIMER0

**SETB** F0

RETI

;——————————延时消除抖动——————————

DLY3**:**

**MOV** R6**,**#50

D1S1**:**

**MOV** R5**,**#50

DJNZ R5**,$**

DJNZ R6**,**D1S1

**RET**

;——————————定时器0延时0.2S——————————

DLY4**:**

**MOV** TMOD**,**#01H ;选择工作模式1，定时器0

**MOV** TH0**,**#3CH ;初值高字节

**MOV** TL0**,**#0B0H ;初值低字节

**MOV** R7**,**#04H

**SETB** ET0 ;定时器T0中断允许

**SETB** TR0 ;启动定时器T0

CLR F0

**RET**

;——————————矩阵键盘、数码管、LED灯码值表——————————

TABLE1**:**

DB 0E7H**,**0EBH**,**0EDH**,**0EEH**,**0D7H**,**0DBH**,**0DDH**,**0DEH**,**0B7H**,**0BBH**,**0BDH**,**0BEH**,**77H**,**7BH**,**7DH**,**7EH

;矩阵键盘：0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

TABLE2**:**

DB 0FCH**,**60H**,**0DAH**,**0F2H**,**66H**,**0B6H**,**0BEH**,**0E0H**,**0FEH**,**0F6H**,**0FAH**,**3EH**,**1AH**,**7AH**,**9EH**,**8EH

;数码管：0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

TABLE3**:**

DB 07FH**,**0BFH**,**0DFH**,**0EFH**,**0F7H**,**0FBH**,**0FDH**,**0FEH**,**07FH**,**0BFH**,**0DFH**,**0EFH**,**0F7H**,**0FBH**,**0FDH**,**0FEH

;对应LED键字0,1,2,3,4,5,6,7,0,1,2,3,4,5,6,7

TABLE4**:**

DB 09EH**,**0EEH**,**0EEH**,**02H**,**0CEH**,**76H**,**0EEH**,**0FCH

;数码管：ERR-PYAO

;歌曲表单

TABLE5**:**

DB 82H**,**01H**,**81H**,**94H**,**84H**,**0B4H**,**0A4H**,**04H

DB 82H**,**01H**,**81H**,**94H**,**84H**,**0C4H**,**0B4H**,**04H

DB 82H**,**01H**,**81H**,**0F4H**,**0D4H**,**0B4H**,**0A4H**,**94H

DB 0E2H**,**01H**,**0E1H**,**0D4H**,**0B4H**,**0C4H**,**0B4H

DB 82H**,**01H**,**81H**,**94H**,**84H**,**0B4H**,**0A4H**,**04H

DB 82H**,**01H**,**81H**,**94H**,**84H**,**0C4H**,**0B4H**,**04H

DB 82H**,**01H**,**81H**,**0F4H**,**0D4H**,**0B4H**,**0A4H**,**94H

DB 0E2H**,**01H**,**0E1H**,**0D4H**,**0B4H**,**0C4H**,**0B4H**,**04H

DB 00H

;生日歌简谱：5 5 6 5 1 7 - | 5 5 6 5 2 1 | 5 5 5 3 1 7 6 | 4 4 3 1 2 1

;音调

TABLE6**:**

DW 64260**,**64400**,**64524**,**64580

DW 64684**,**64777**,**64820**,**64898

DW 64968**,**65030**,**65058**,**65110

DW 65157**,**65178**,**65217

;低5，低6，低7，中1

;中2，中3，中4，中5

;中6，中7，高1，高2

;高3，高4，高5

END

## 四、结果分析

1.功能1按键显示0~F可以顺利实现，如图所示。

2.功能2按键控制相应LED灯点亮顺利实现，结果如图。

3.中断0实现数码管ERR-闪烁3次，我运用了定时器，理论应该亮1s，暗0.6s，但事与愿违，时间比计划短，如图所示。

4.中断1实现走马灯10次，顺利完成，结果如图所示。

5.矩阵键盘与蜂鸣器可以实现音乐键盘功能，结果可以倾听生日歌，但是由于蜂鸣器质量原因，音色比较差，换同学焊接的喇叭之后音色有很大改善。

**输入按键并显示**：



图4.1当键入按键，在一位数码管上显示键值0



图4.2当键入按键，在一位数码管上显示键值1



图4.3当键入按键，在一位数码管上显示键值2

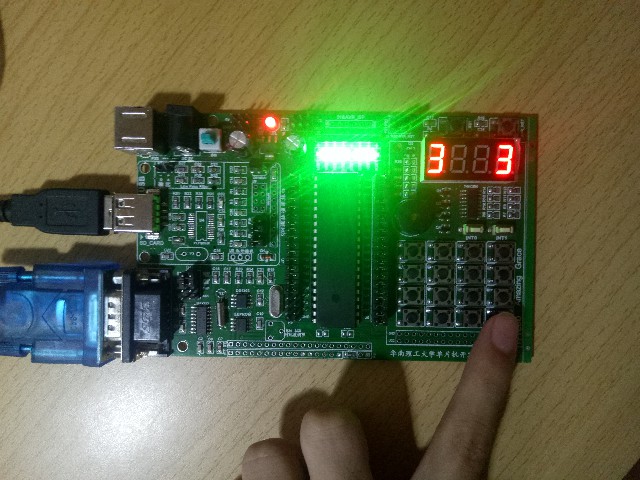


图4.4当键入按键，在一位数码管上显示键值3



图4.5当键入按键，在一位数码管上显示键值c



图4.6当键入按键,在一位数码管上显示键值d



图4.7当键入按键，在一位数码管上显示键值E



图4.8当键入按键，在一位数码管上显示键值F

**键盘控制LED:**

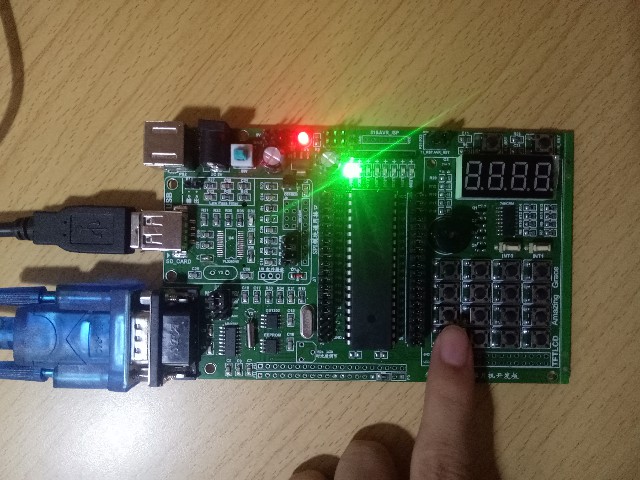


图4.9当键入其中0按键，点亮相应的LED。

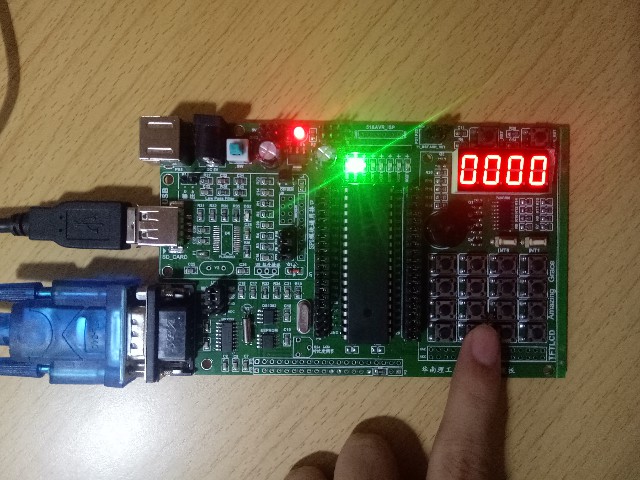


图4.10当键入其中1按键，点亮相应的LED。

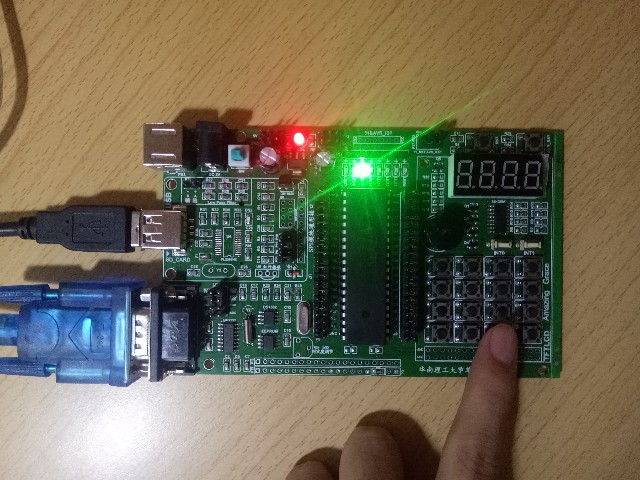


图4.11当键入其中2按键，点亮相应的LED。

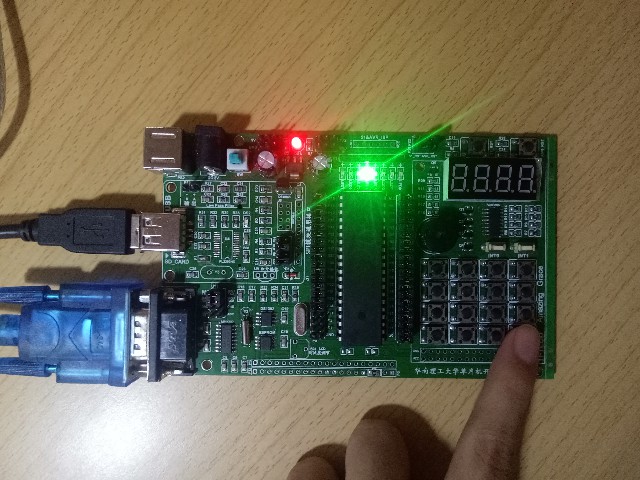


图4.12当键入其中3按键，点亮相应的LED。

**外部中断0：**



图4.13按下KEY1，在四位数码管闪烁显示“ERR-”三次

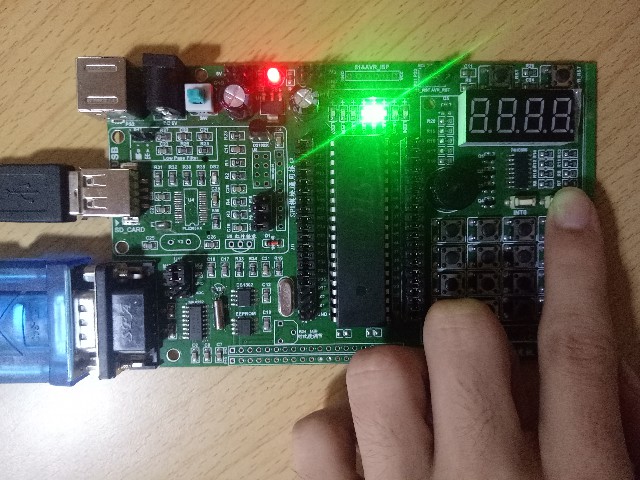
**外部中断1：**

图4.14按下KEY2，LED演示走马灯10次

## 五、课程设计总结

**收获：**通过本次单片机课程设计，我对中断的实现、定时器的定时、按键的扫描以及数码管的显示等等有了比较深入的了解。

**问题：**刚开始时，编译软件、烧录软件的使用，串口的使用很有疑惑，通过网上搜索，最终学会了将stc添加到keil的device里，串口也可以通过软件扫描识别，keil也可以编译导出hex文件直接用于烧录了。打代码时，由于还没学到键盘扫描，所以只能先自学，通过学习课本，了解到3种方法，比较而知反极法最优秀，所以选了这种方法。然后对74HC595的传输方式不太了解，又搜索学习了该芯片，知道了串口输入，并口输出等。接着了解了中断的入口地址，子程序的编辑等。最后复习又重新学习了定时器0,1的一些设置。对51单片机深入了解可以说大有收获。

**思考：**这次课设收获很大，从一无所知到熟悉掌握，可以将课堂所学用于实践，提升自我的同时感受到了学习的快乐与成就感。希望老师可以给我们单独留段时间进行学习与拓展，相信将会实现更加完美优秀的功能。本次经过自己的刻苦努力将题目要求全部实现，并综合到一起，还经过学习课本后面章节实现了播放生日歌音乐的功能，自我评价优秀。

**参考文献**

**[1]林土胜.单片机技术及工程实践.北京：机械工业出版社，2011.68~163**