# 微

# 机

# 原

# 理

# 实

# 验

# 报

# 告

# 学院：电子信息与通信学院

# 姓名：周睿科 赵展

# 学号：U202113957 U202117282

# 实验时间：2023年7月11日

1. **实验名称**

走马灯动态演示实验

1. **实验任务**

（基础部分）

* 1. 用可重定位汇编语言实现，做出流程图
  2. 实现数字译码和动态显示（每位数码管不同内容）
  3. 实现数码管显示的滚动效果（走马灯）

（拓展部分）

1. 用数码管呈现实验三外中断部分的效果
2. 构建良好的显示模块，使显示部分易于扩展
3. 提供更多的显示控制，如闪烁、小数点
4. 提供更多的译码组合，制作开机画面
5. **实验条件**
   1. MPLAB X IDE v6.10（代码编写与仿真）
   2. Xc8 编译器
   3. PIC16F18854 单片机及一些拓展模块
   4. 面包板、0.28 英寸极性共阴四位数码管、电阻等元器件
6. **实验设计方案——硬件**
   1. **电路设计：**

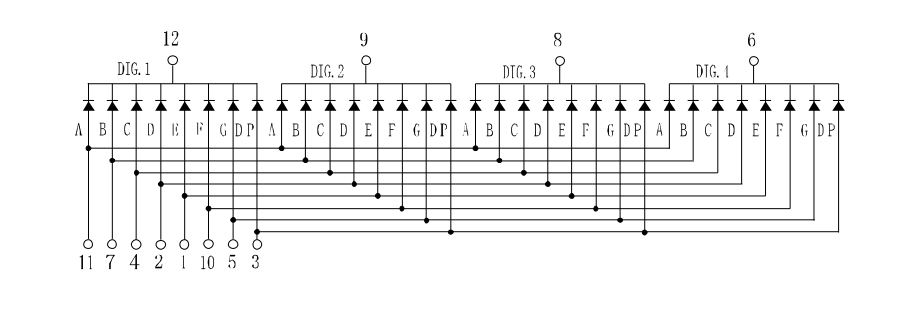


图 1 数码管控制引脚

如图一电路连接采用PORTA0~6位连接A~G，PORTC0~3位对应位码控制数码管1~4的亮灭。

1. **实验设计方案——软件**
   1. **设备配置**
      1. PORTA：TRISA = 00000000全端口输出
      2. TIMER0：
         1. T0CON0：10000000B 使能Timer0，8bit模式，1:1分频
         2. T0CON1：01010000B 时钟FOSC/4，1:1分频
      3. 中断相关寄存器设置：
         1. BCF PIR0, 5
         2. BSF INTCON, 6
         3. BSF INTCON, 7
         4. BSF PIE0, 5
   2. **设计思路**
      1. 走马灯的切换通过主程序来间接控制：

* 直接延时，计数结束后，Flag变量左移。
* 在ISR中，Flag具有类似掩码的效果。
  + 1. 直接把所有移动的数字存放在不同的查找表中，一个查找表对应 Flag 全局变量中的一bit
    2. 软件主要采用中断 + 置数的模式，核心有三个变量：
* segOffset：段码偏移，通过该变量实现查表操作
* posCode（output）： 位码，通过移位指令实现扫描。低四位输出。
* segCode（output）： 段码，八位均输出。
  + 1. 具体各个变量解释：
* 所有变量同步置数
* segOffset：从 0 开始，每次循环 + 1。当其为 4 时，返回并重新置数。
* posCode：从11110111开始循环右移，高位补0，但不影响低位的输出。
* segCode：具体内容存在查找表中。
  1. **软件流程图**

|  |  |
| --- | --- |
| 图 2 主程序流程图 | 图 3 ISR流程图 |

* 1. **主程序流程解释**
     1. 首先配置IO端口及TIMER0计时器
     2. 计数器中断使能
     3. 配置Flag，初始为0x01，Flag不断左移，设置循环，通过延时函数控制数码管的滚动效果
     4. 若Flag的第四位等于1，则说明循环了四次，应重新reset Flag为0x01
     5. 若Flag的第四位不等于1，跳过reset指令
  2. **ISR流程解释**
     1. 首先重置计时器
     2. 右移pcode控制选择一个数码管亮
     3. 通过Flag选择不同的table
     4. 根据segoffset的值选择不同的数字显示
     5. 判断segoffset的值是否达到4，达到了就reset

1. **实验源代码及注释**

intentry:

; Interrupt service routine

; clear overflow flag

banksel PIR0

bcf PIR0, 5

; output:

call func\_output

;right shift position code

rrf pCode, f ;位码

;use offset to search in table

;incf segOffset, f

movf segOffset, w

btfsc Flag, 0

call Table\_1

btfsc Flag, 1

call Table\_2

btfsc Flag, 2

call Table\_3

btfsc Flag, 3

call Table\_4

incf segOffset, f

movwf segCode

btfss segOffset, 2

retfie

\_reset:

call func\_delay\_op

call func\_output

; reset vars

movlw 11110111B

movwf pCode

movlw 0

movwf segOffset

retfie

psect main,class=CODE,delta=2 ; PIC10/12/16

global \_main

\_main:

;TODO

;PORTA config

banksel PORTA

clrf PORTA

banksel LATA

clrf LATA

banksel ANSELA

clrf ANSELA

banksel TRISA

movlw 00000000B

movwf TRISA

;PORTB config

banksel PORTC

clrf PORTC

banksel LATC

clrf LATC

banksel ANSELC

clrf ANSELC

banksel TRISC

movlw 00000000B

movwf TRISC

;TIMER0 config

banksel T0CON0

movlw 10000000B

movwf T0CON0

banksel T0CON1

movlw 01010000B

movwf T0CON1

banksel PIR0

bcf PIR0, 5

banksel TMR0H

movlw 0x7f

movwf TMR0H

; INTERRUPT config

banksel PIR0

bcf PIR0, 5

; GIE, PIE enable

banksel INTCON

bsf INTCON, 7

bsf INTCON, 6

; TMR0IE enable

banksel PIE0

bsf PIE0, 5

; go!

call set\_pcode

Flag equ 0xf7

movlw 0x01

movwf Flag

loop:

btfsc Flag, 4 ;第四位为1跳过下一条指令

call \_reset\_flag

call func\_delay

rlf Flag ;不断左移

goto loop

\_reset\_flag:

movlw 0x01

movwf Flag

return

set\_pcode:

movlw 11110111B

movwf pCode

movlw 1

movwf segOffset

movlw 00000110B

movwf segCode

return

Table\_1:; Attention: RA7 == RA6

addwf PCL, f

retlw 00000110B ;1

retlw 11011011B ;2

retlw 11001111B ;3

retlw 11100110B ;4

Table\_2:; Attention: RA7 == RA6

addwf PCL, f

retlw 11100110B ;4

retlw 00000110B ;1

retlw 11011011B ;2

retlw 11001111B ;3

Table\_3:; Attention: RA7 == RA6

addwf PCL, f

retlw 11001111B ;3

retlw 11100110B ;4

retlw 00000110B ;1

retlw 11011011B ;2

Table\_4:; Attention: RA7 == RA6

addwf PCL, f

retlw 11011011B ;2

retlw 11001111B ;3

retlw 11100110B ;4

retlw 00000110B ;1

func\_delay\_op:

count1\_op equ 0x22

banksel 0

movlw 10

movwf count1\_op

delay\_1:

decfsz count1\_op, 1

goto delay\_1

return

func\_delay:

count1 equ 0x20

count2 equ 0x21

banksel 0

movlw 200

movwf count1

delay\_1\_op:

movlw 200

movwf count2

delay\_2\_op:

decfsz count2, 1 ;count2值减一

goto delay\_2\_op

decfsz count1, 1

goto delay\_1\_op

return

func\_output:

; TODO

; segment

banksel PORTA

movf segCode, w

movwf PORTA

; position

banksel PORTC

movf pCode, w

movwf PORTC

return

; END

exit:

nop

;END

end reset\_vec