**单片机原理与接口技术实验**

学 院 自动化学院

专 业 电子信息科学与技术

年级班别 2017 级 1 班

学 号 3117001295

学生姓名 方智威

指导教师 王倩雪

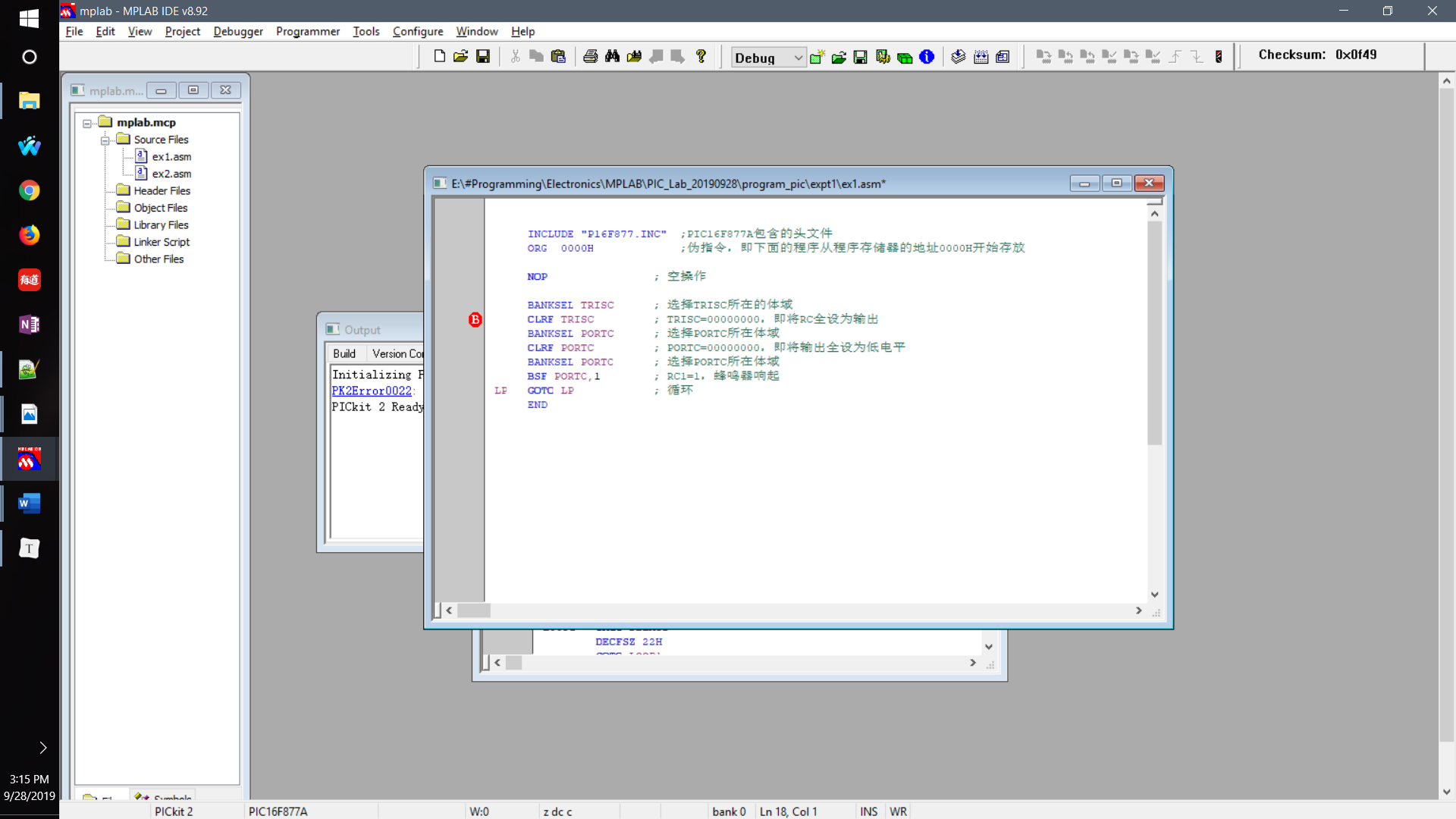
2019年 9 月28日

实验一

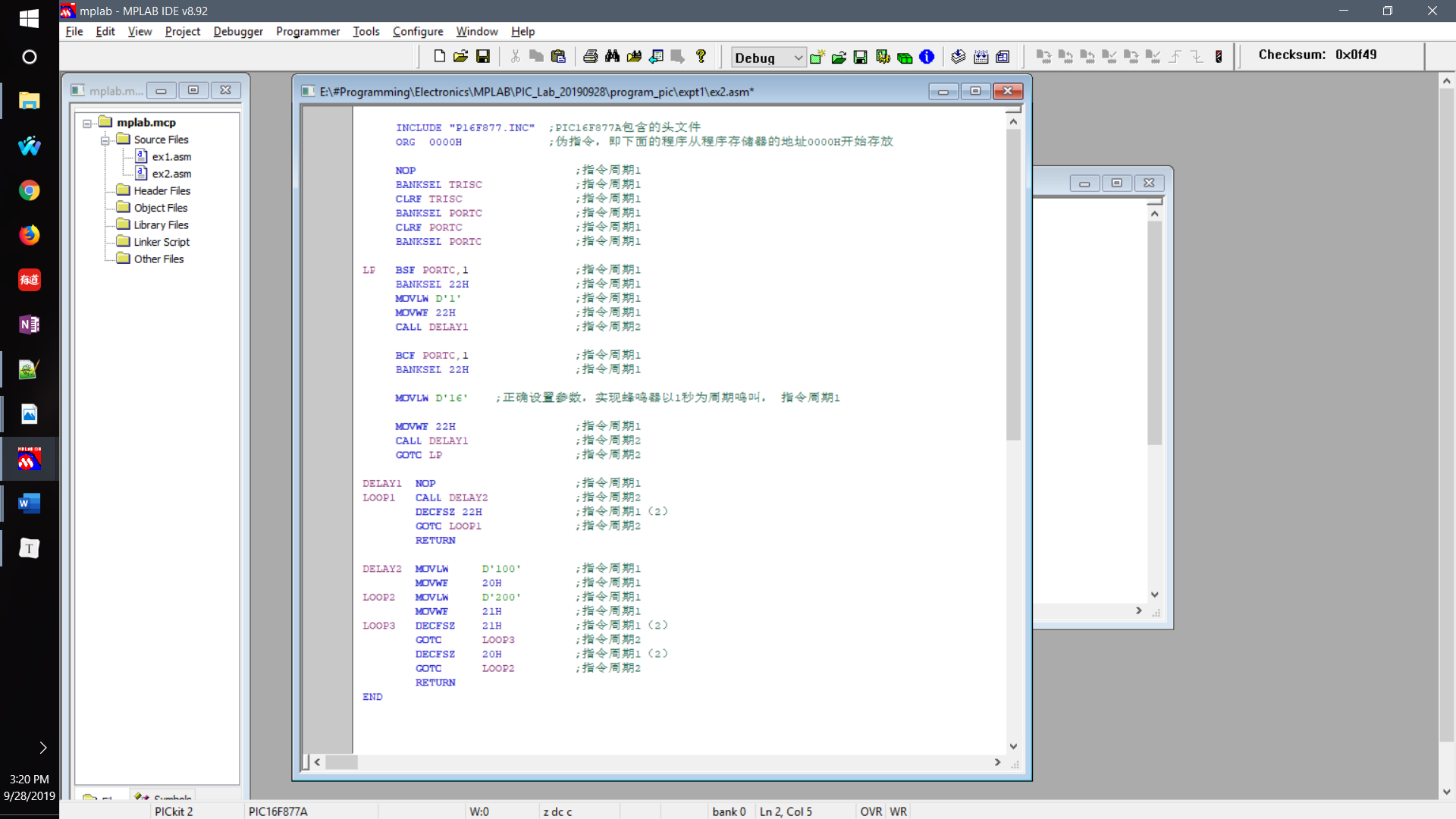
**实验项目名称：PIC单片机汇编语言编程及硬件电路调试**

## 一、报告内容

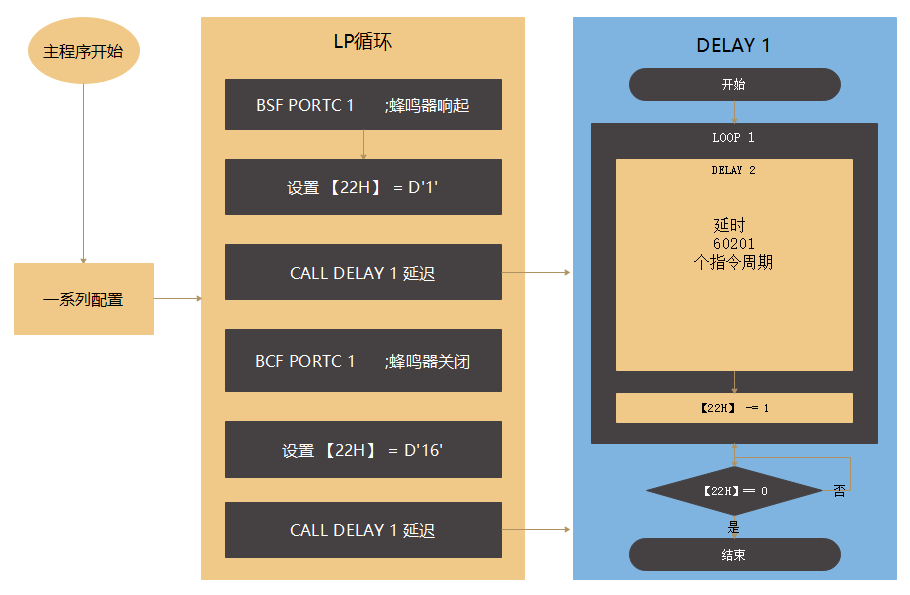
1、对例程1的每一句指令进行注释，描述该句指令的作用以及如何影响寄存器结果。



2、标出例程2每一句指令的指令周期，伪指令除外。



3、写出例程2的程序流程图。



4、写出方框内参数的计算过程。

DELAY2 = 1+1【 (【(1+2)x200 – 1】+ 1 + 2 ) x 100 – 1】 = 60201

DELAY1 = (【DELAY2】+1+2) \*【22H】 = 1000000

解得 【22H】= 16.610192

## 二、思考题

待补

## 三、实验中遇到的问题和解决途径

问题一：计算指令周期，在多个循环时容易出错混乱

解决：仔细看代码，逐行剖析

实验二

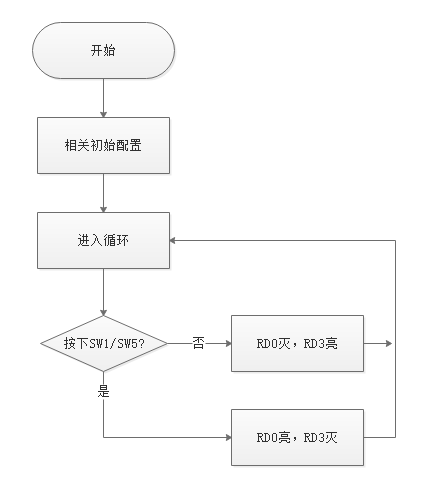
**实验项目名称：PICC 编译环境及PIC单片机的通用IO口编程**

**一、报告要求**

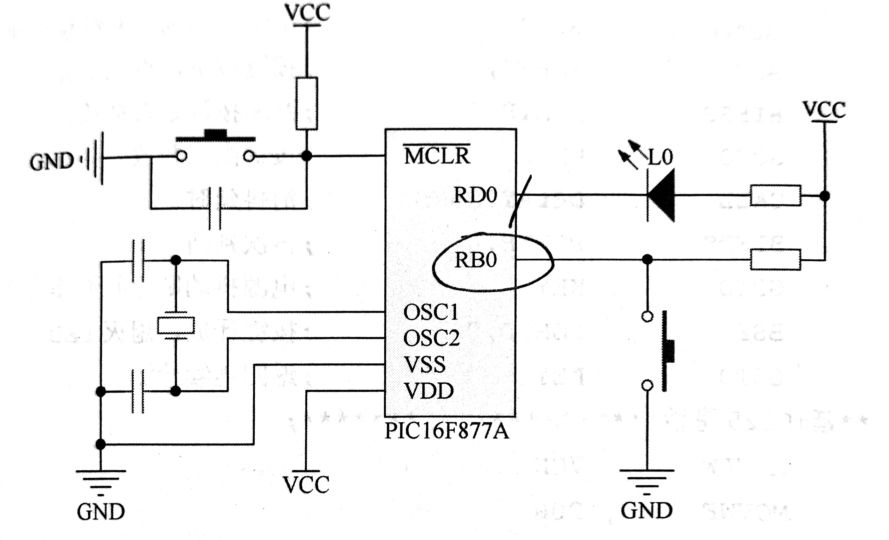
1. 对例程1的每一句程序进行注释。描述如何改变寄存器及其对IO的影响。



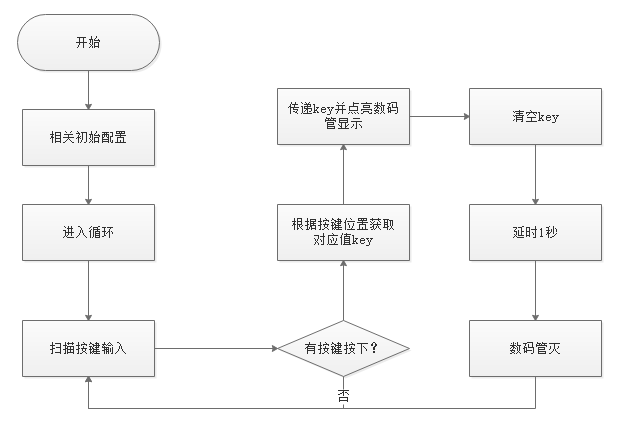
2. 画出例程1的程序流程图。



3. 画出例程2所涉及的硬件电路图，要求包括最小系统，IO口，LED，键盘电路。



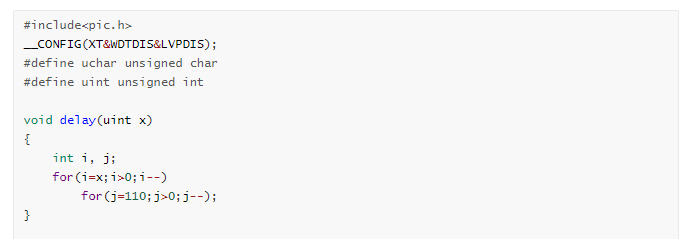
4. 画出例程2的程序流程图，包括主程序及子程序。

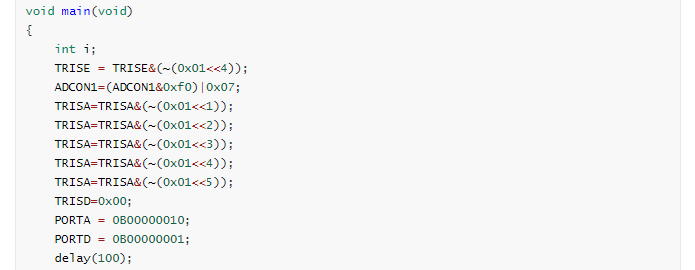


5.讨论红色表及部分的delaytime 的作用以及其参数的影响。

因为电路设计的原因，多个数码管共用PORTD 数据口，所以正常情况下只能多个数码管显示同一个值。为能充分利用数码管，我们可以设置极短延时让不同数据在不同数码管间快速切换，达到一种像是显示着不同数值的现象。就好像该题的 delaytime 延时，若 delaytime 过长，则会导致数码管与数据的切换过慢，导致两个数码管来回闪烁而不是同时显示，达不到我们想要的效果。因此我们需要这个 delaytime 尽可能的短但不可以没有

## 二、思考题







## 三、实验中遇到的问题和解决途径

问题一：计算指令周期，在多个循环时容易出错混乱

解决：仔细看代码，逐行剖析

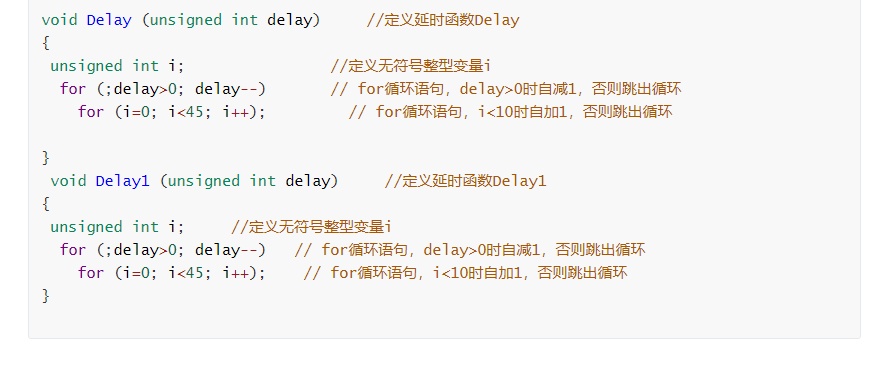
实验三

**实验项目名称：PIC单片机中断实验**

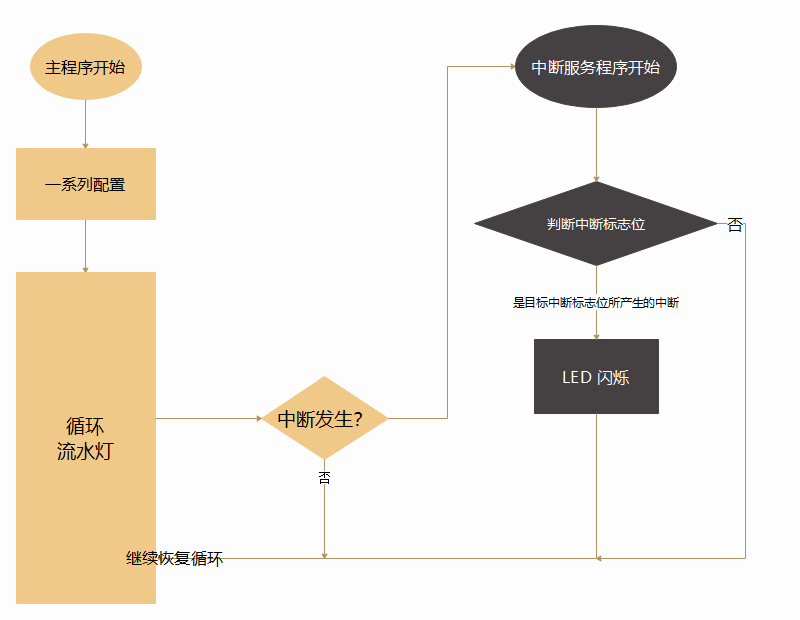
**一、报告要求**

1. 对例程1的每一句程序进行注释。描述如何改变寄存器及其对IO的影响。





2. 画出例程1的程序流程图。



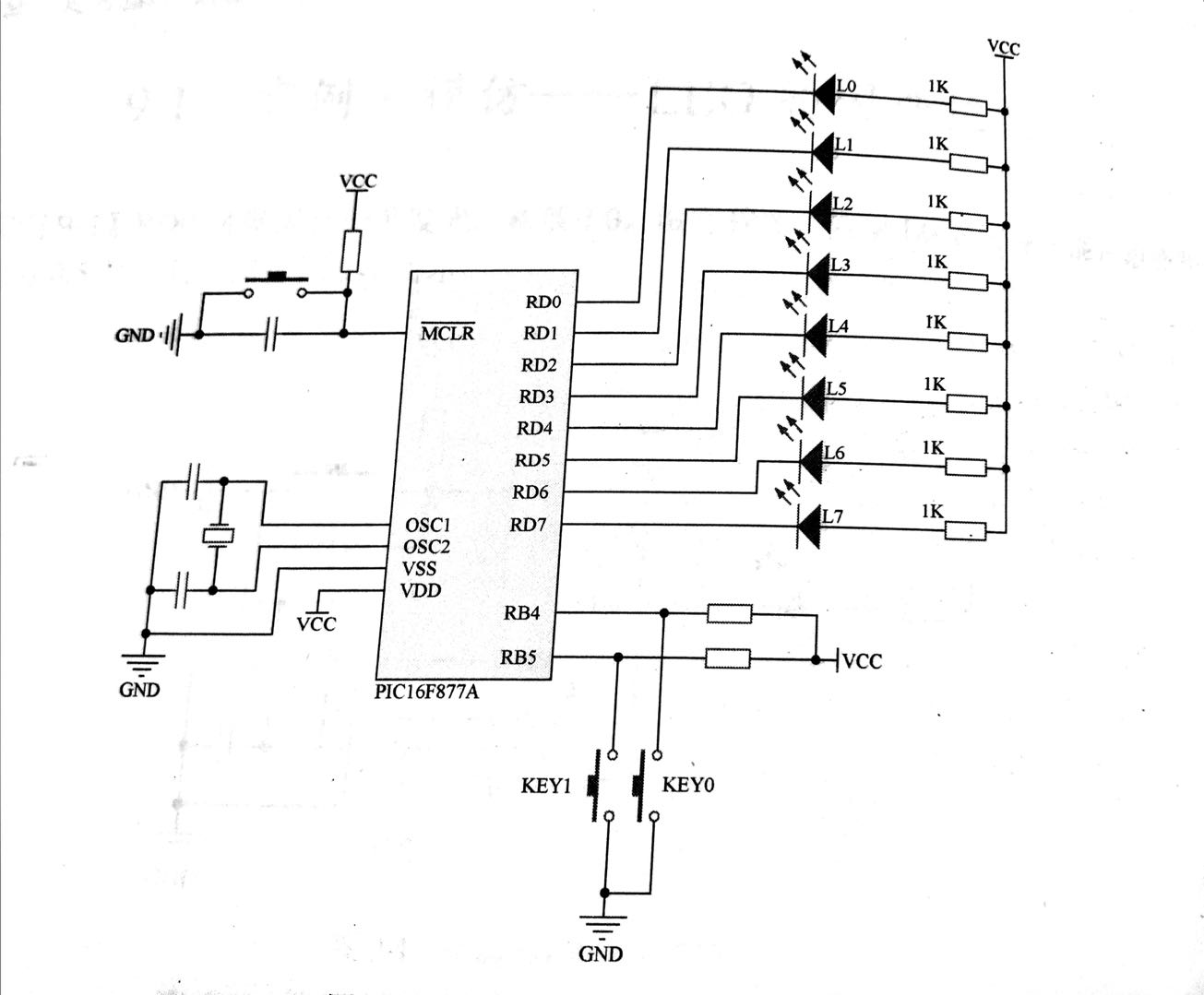
3. 讨论中断方式与查询方式的区别。

通过查询方式进入中断服务程序，则需要在程序中不断去扫描标志位。而通过中断方式来执行中断服务程序，能避免一些持续扫描而造成不必要的资源浪费，而且在任何时候只要中断标志位更改，能及时跳转的中断服务程序，不像查询方式有相对延时，需要等到扫描步骤执行才能去判断中断是否触发。

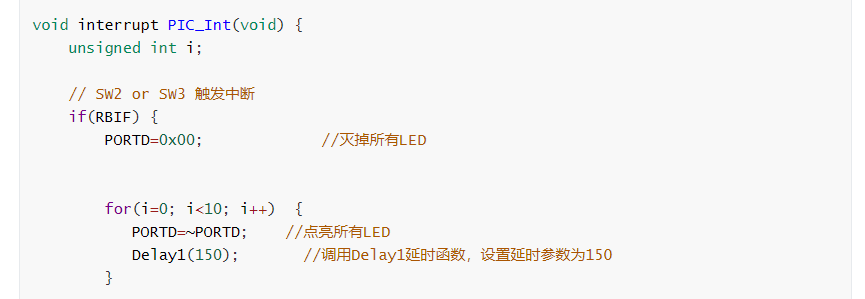
4. 讨论 INTEDG的作用和对INT中断触发的效果。

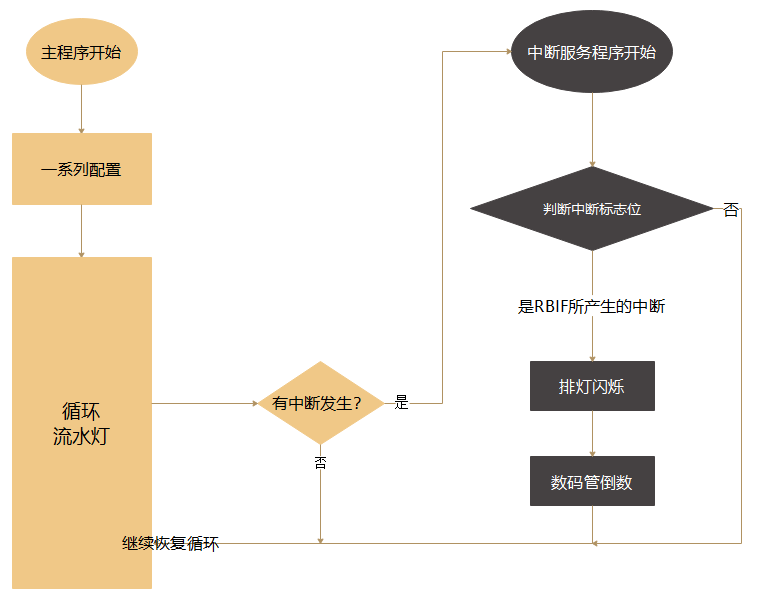
INTEDG 位是用来配置 RB0/INT 引脚输入中断信号是上升沿触发还是下降沿触发，若 INTEDG = 1，则 RB0的信号为上升沿时触发中断；若 INTEDG = 0，则 RB0的信号为下降沿时触发中断。

5. 画出任务2所涉及到的硬件电路图，要求包括最小系统，IO口，LED，相关键盘电路。



6. 编写出任务2的中断服务程序以及程序流程图。

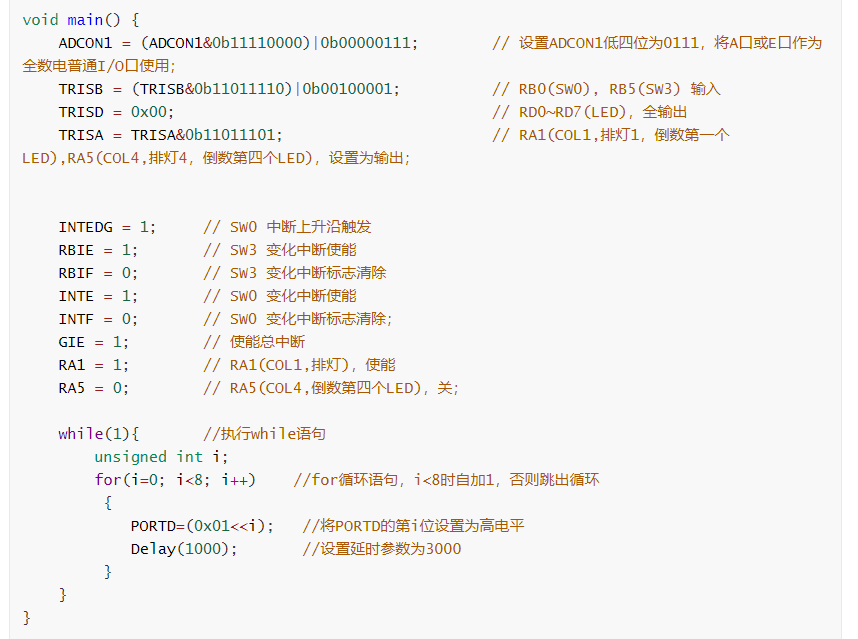
 



**二、思考题**

编写中断服务程序，实现当SW3按下时，触发RB电平变化中断，LED数码管第四位倒数5,4,3,2,1,0，当SW0 按下时，触发 INT 中断，第一列 LED 闪烁 5 次。









## 三、实验中遇到的问题和解决途径

问题一：思考题中两种不同触发中断下的中断清除会有问题。

解决：把中断标志位清除放在判断语句外。

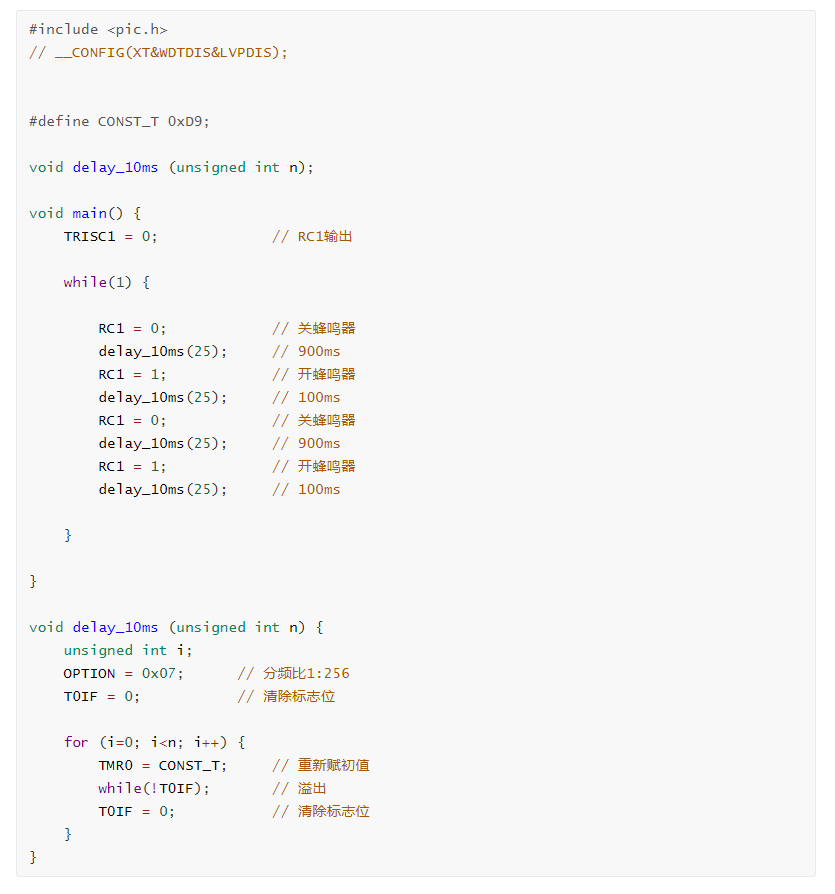
实验四

**实验项目名称：PIC单片机定时器实验**

**一、报告要求**

1. 对例程1、2、3的每一句程序进行注释。描述如何改变寄存器及其对IO的影响。

* 例一



* 例二



* 例三



2，详细写出任务 1 中定时器初值的计算过程。

选用12M晶振

256\*（256-Y）\*（4/12）=10000

解得 Y=139=0x8B

3，画出例程 2 的程序流程图。

开始-相关配置初始化

LED0亮

j=j+1

LEDj-1灭，LEDj亮

调用延时程序delay（500）

No

j>7？

Yes

产生中断，蜂鸣器鸣叫

清除中断标志位

重新赋予定时器TMR0初值

4，讨论定时器查询方式和中断方式的差异和优缺点。

中断方式会自动对中断标志进行扫描和捕获，而查询则需自己实现对中断标志的扫描。

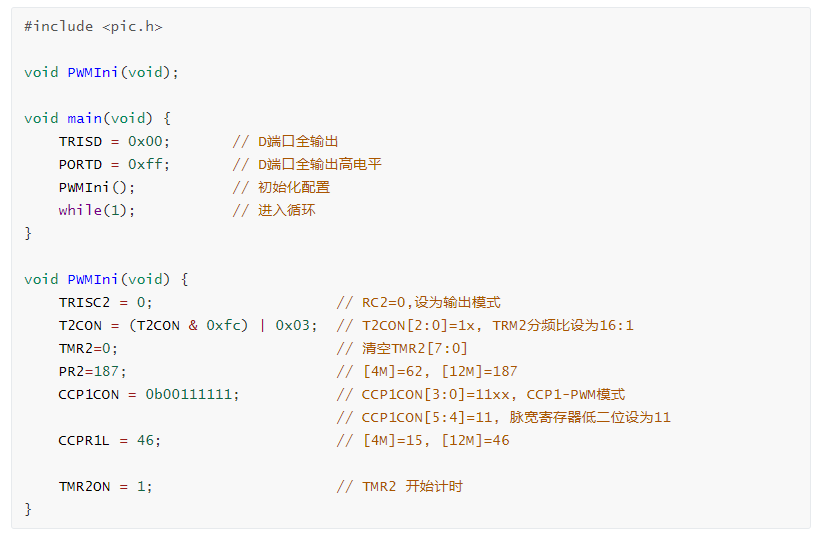
中断方式的优点在于，能较为实时地捕获到中断标志位，而不需要像查询方式那样需要等到查询程序的开始才能进行中断捕获。

实验五

**实验项目名称：PIC单片机PWM实验**

**一、报告要求**

1，对例程 1 的每一句程序进行注释。描述如何改变寄存器及其作用。



2，详细写出任务 1 中定时器初值的计算过程。

【4M晶振, 1KHZ, 25%占空比】

1x10^(-3) = 4x(1/4)x10^(-6) x 16 x (1+PR2)

1x10^(-3) x25%= 4x(1/4)x10^(-6) x 16 x CCPR1L

解得 PR2 = 62, CCPR1L = 15

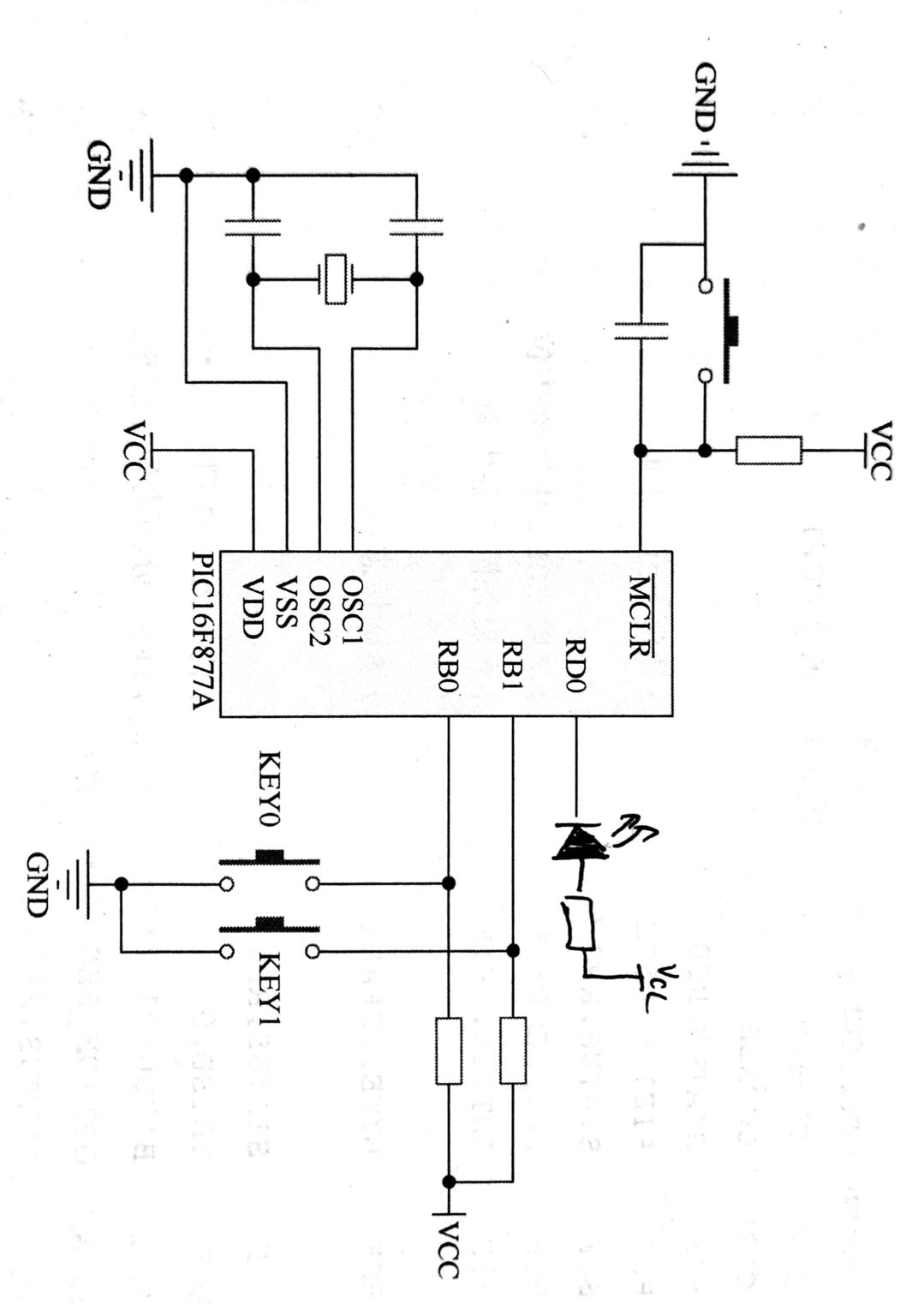
【12M晶振, 1KHZ, 25%占空比】

1x10^(-3) = 4x(1/12)x10^(-6) x 16 x (1+PR2)

1x10^(-3) x25%= 4x(1/12)x10^(-6) x 16 x CCPR1L

解得 PR2 = 187, CCPR1L =46

3，画出任务 2 的电路图。



4，画出任务 2 的程序流程图。

把端口D设置为输出

把端口D设置为输出高电平

调用PWMIni（）

RB4=0?

RB2=0?

Yes

Yes

调用Delay（）

调用Delay（）

RB4=0?

RB2=0?

Yes

Yes

CCPR1L>?

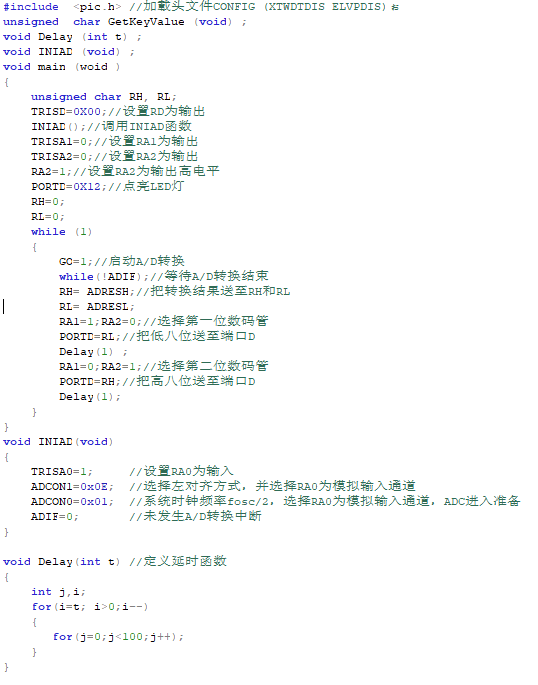
CCPR1L<PR2?

实验六

**实验项目名称：PIC单片机AD采样实验**

**一、报告要求**

1，对例程 1 的每一句程序进行注释。



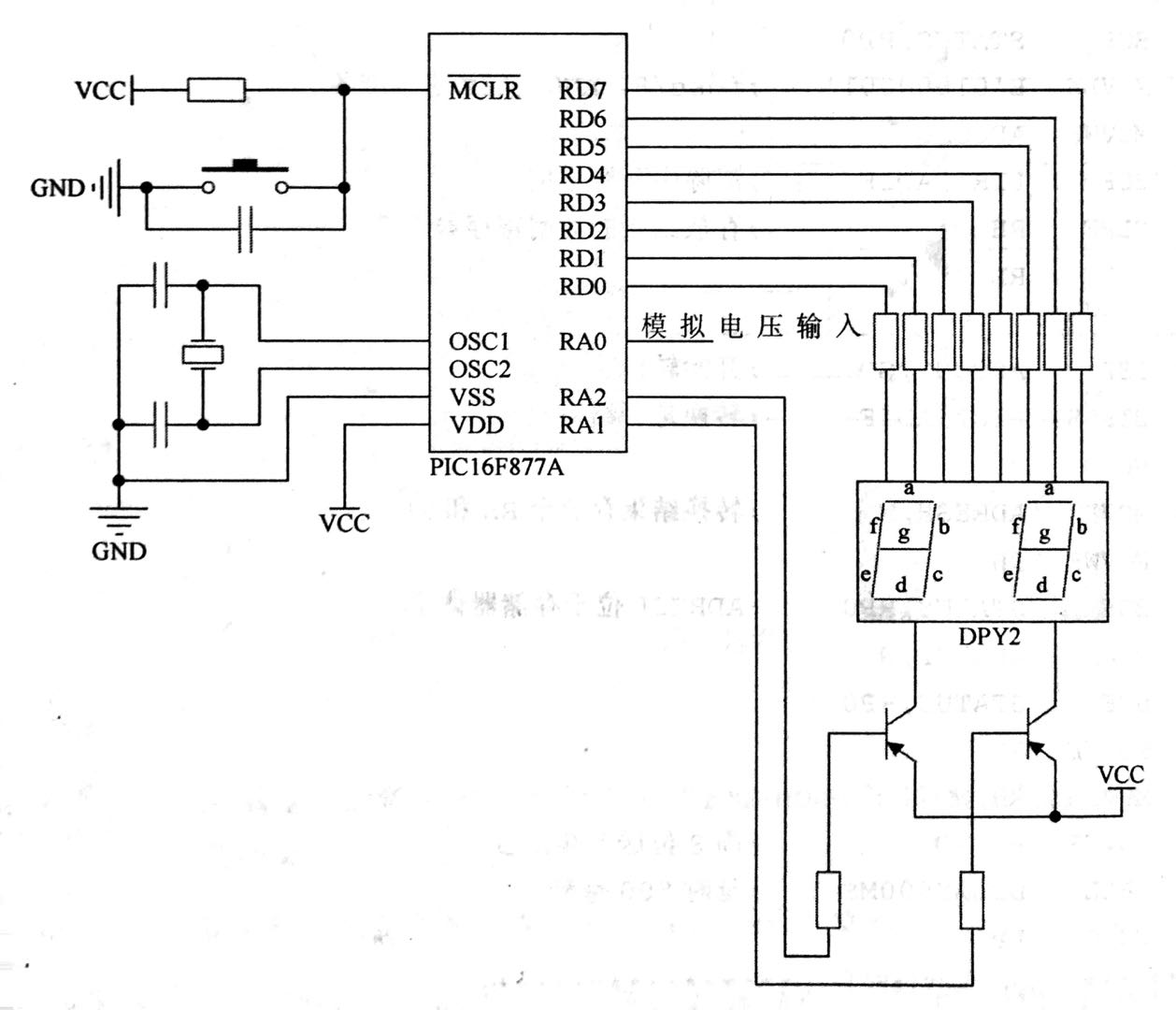
2，写出例程 1 中，对 AD 相关寄存器进行初始化的值（括号处）。并解释为什么这样设置。

GO=1时要启动A/D转换，

ADCON1=0x0E 是要选择左对齐方式，并选择RA0为模拟输入通道

ADCON0=0x01是要选择系统时钟频率fosc/2，选择RA0为模拟输入通道，ADC进入准备

3，画出 2 的硬件电路图。



4，画出任务 2 的程序流程图，包括主程序，子程序。

INIAD( )

定义变量AD\_result

将RA0设置成输入

将RA1,RA2,RA3,RA5,RD设置成输出

调用INIAD( )

定义变量a,b,c,d

启动AD转换

选择右对齐方式，并选择RA0/AN0为模拟输入通道

将AD采样结果存放在AD\_result

选择频率focs/2 并开启AD转换

将AD\_result的千十百个位的值分别赋给a,b,c,d

将a,b,c,d的值显示在数码管上和LED上

清除标志位

实验七、八

**实验项目名称：PIC单片机串行通信**

**一、报告要求**

1，填空完成例程 1。



2，写出成功通信的波特率值以及其对应的 SPBRG 值。

【4M晶振-高速挡】

9600 = (1/4)x10^(-6)/[16x(SPBRG+1)]

解得 SPBRG = 25

3，写出任务 3 程序流程图，并贴出程序

调用INIUSART（）程序

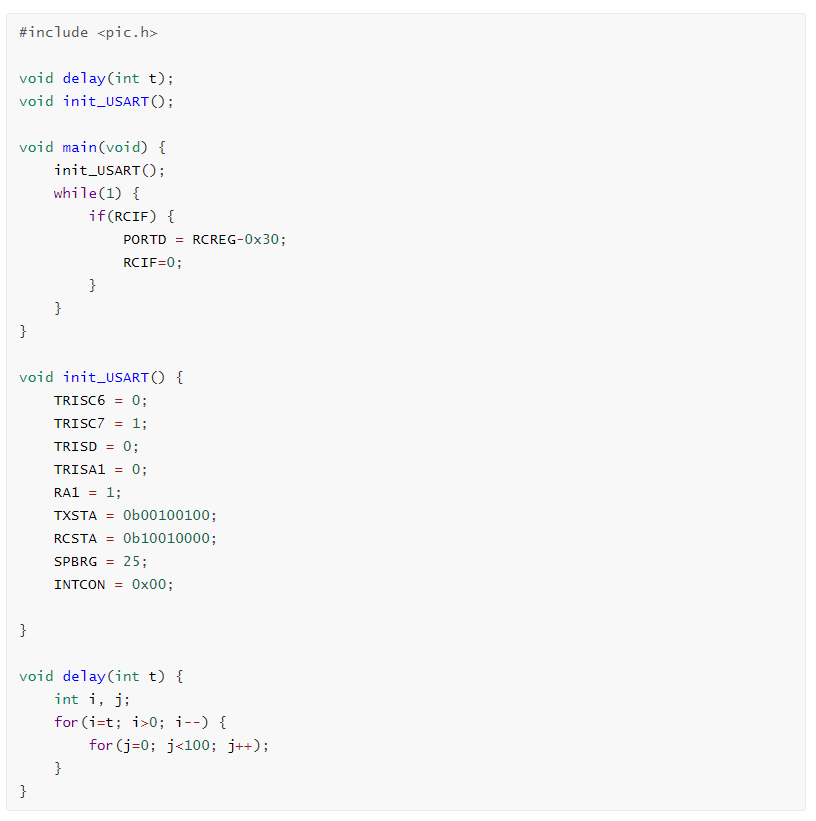
把端口A和D设置为输出

发送数据“2”到PC端

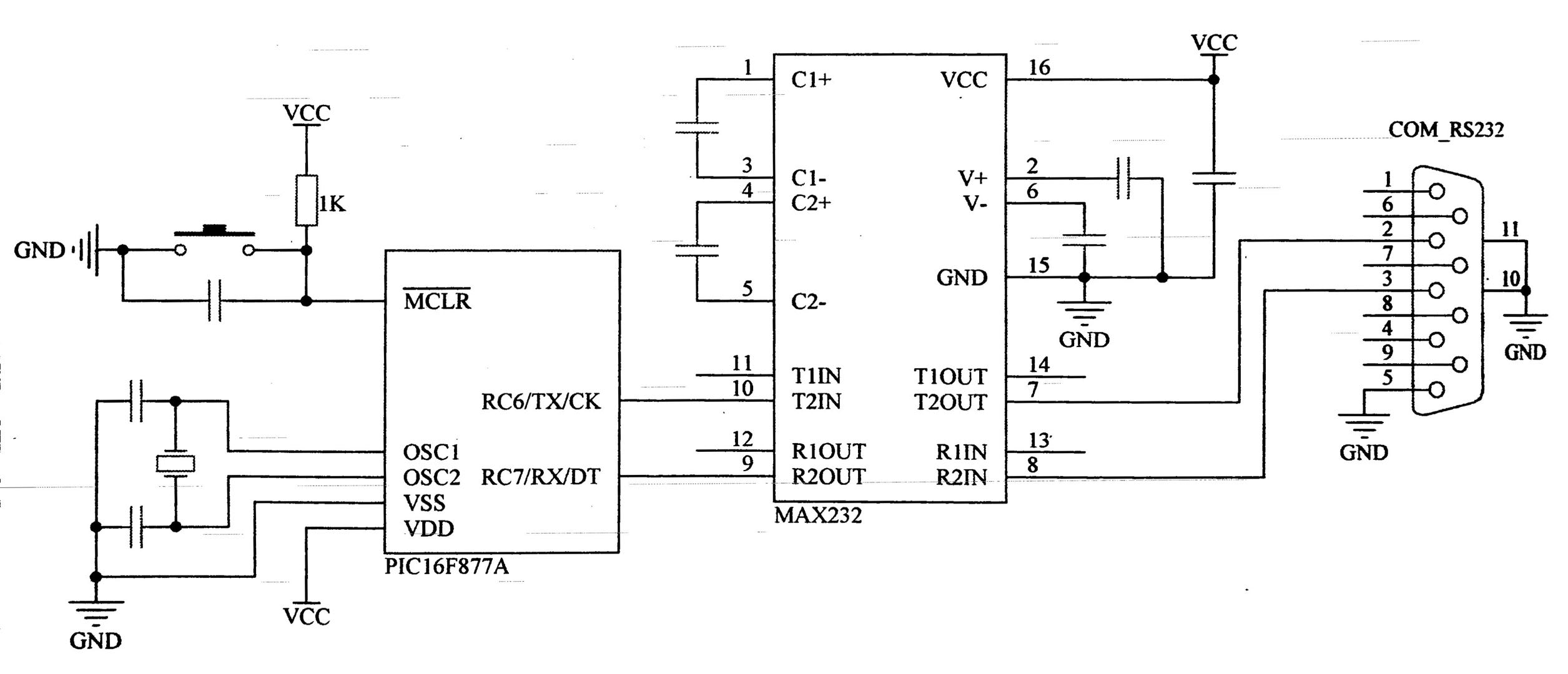
等待发送完成

发送完成

数码管显示“2”



4，画出任务 4 的硬件电路图及程序流程图，包括主程序，子程序。







定义变量AD\_result

调用INIAD( )

调用INIUSART( )

定义变量a,b,c,d

启动AD转换

将AD采样结果存放在AD\_result

将AD\_result的千十百个位的值分别赋给a,b,c,d

将a,b,c,d的值按序发送到PC端

其显示的正是AD采样的值

将a,b,c,d的值显示在数码管上和LED上

INIAD( )

INIUSART( )

将RA0设置成输入

将RA1,RA2,RA3,RA5,RD设置成输出

把RC6设置为输出

把RC7设置为输入

选择异步方式 高速波特率

发送使能 8位数据

选择右对齐方式，并选择RA0/AN0为模拟输入通道

使能串口端口功能，

连续接收数据

选择频率focs/2 并开启AD转换

禁止中断

清除标志位