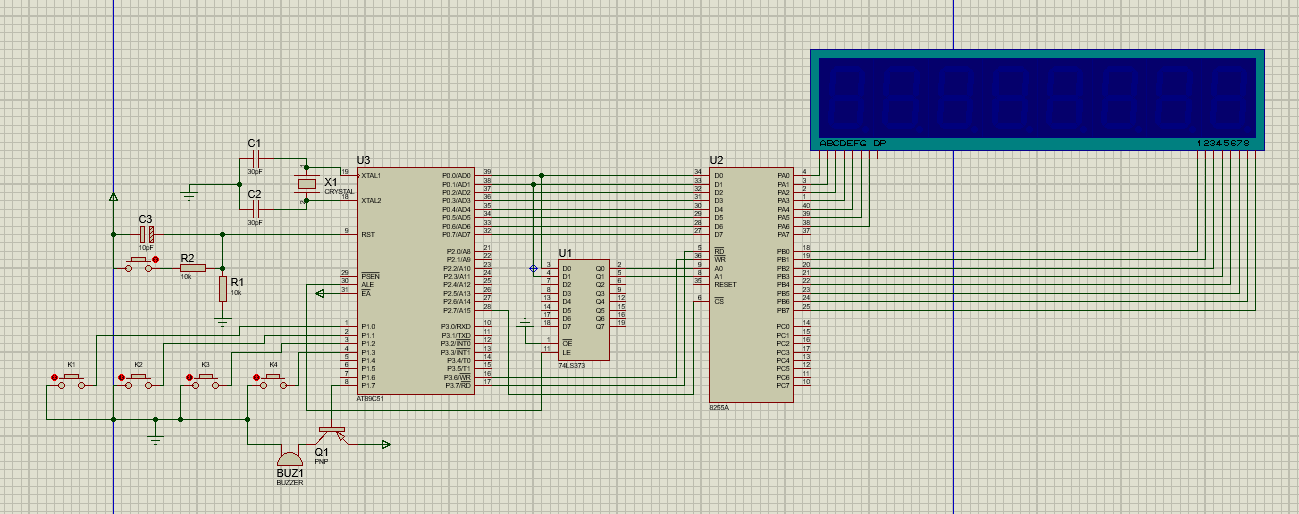
基于51单片机的电子钟

**一:电路设计**

 电路图如下:

该电路包括单片机的最小系统,74ls373芯片,以及用于扩展IO的8255芯片,一个有源蜂鸣器,一个8连管,以及四个按键组成。

**1):引脚连接:**

按键分别接在单片机的P10到 P13口,有源蜂鸣器由P17引脚通过PNP的三级管进行对蜂鸣器是否发声进行控制,74ls373在单片机与8255直接起到过渡作用,8255的数据线连接到单片机的P0口,8连管在8255的PA 以及 PB口连接,PA口控制显示的数值,PB口进行显示的选择。

按键以独立按键的形式进行控制,使用独立按键有利于简化程序的编写,而且4个按键的话矩阵按键与独立按键的IO口消耗数量相同,所以选择的独立按键的连接方式。对于蜂鸣器的选择,有源蜂鸣器在正负极之间只要有电压差到达一定条件蜂鸣器就会发声,控制起来相对比较简单,使用无源蜂鸣器的话需要一个不断输出PWM的引脚来进行蜂鸣器的发声控制,编程上会多一些操作,因此直接选择了有源蜂鸣器进行时间到的提示器件,在设置时间到后,蜂鸣器发声进行提示。

硬件的链接方式。连接方式如下：按键连接在P1^0~1^3端口，一共四个，起到控制作用；蜂鸣器接在P1^7 用于提醒时间到了；8255IO口拓展芯片，接在P0端口扩展IO口；8连管用于显示数字，接在8255的输出引脚；74ls373起到过渡作用

**功能实现总体思路：**

仿真开始在数码管上动态显示时间，7秒后切换到日期显示，在显示的过程中使用K1进行时间与日期的手动切换功能，在切换后保持10秒，之后恢复之前的交替显示的状态，使用K2进行12/24小时的切换操作，每按下一次进行一次12/24小时显示的切换操作，切换后从当前按下时即起作用，即在显示时间的时候按下K2，则时间立即切换为24小时显示的状态，如果当前显示的是日期则，下一次切换到时间显示的时候显示时间的24小时形式。

初始设置时间以及日期在程序中固定，可以使用K3进行调整，当K3按键按下即进入时间日期调整模式，在调整过程中K1 K2 K4键不在是之前的按键功能，而是对应的对数值的修改的操作。闹钟功能的初始时间设置为程序中设定，在修改时间日期的功能下能够进行对定时时间的修改，在倒计时结束后，蜂鸣器会进行发声提示操做，而且在修改的过程中倒计时，正常时间计数都会停止而且正在修改的位置数字会进行闪烁显示，用于提示当前的修改位置，使用K4按键能够进行闹钟的开启与关闭操做，在闹钟开启的模式下8连管的最后的小数点位置会被点亮来提示当前闹钟在开启状态，在闹钟倒计时结束时可以通过任意按键停止蜂鸣器的发声，并且将计时恢复之前的状态，继续进行倒计时，除非使用K4键关闭闹钟，并且在没有操作的30秒后，闹钟计数也会恢复之前的状态。

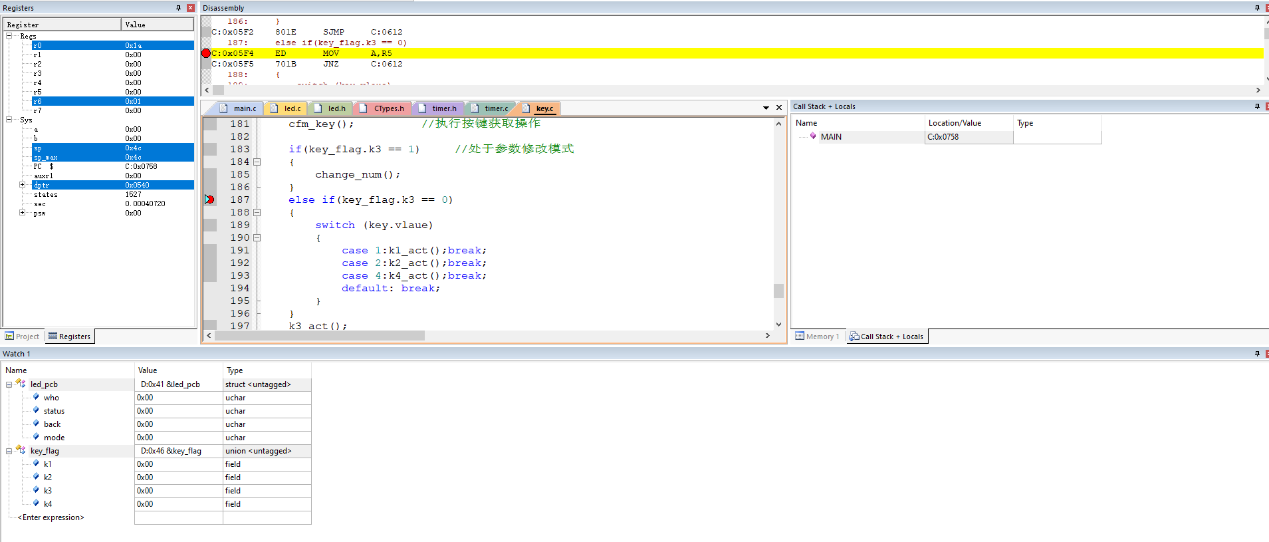
**3）：在理清楚需要的功能之后就可以绘制整个程序的流程图方便对整个工程进行有序的整理以及编写。**

**4）：调试过程：**

在keil 5中直接进行软件的仿真测试，在没有错误的情况下将hex导入到Proteus中进行在电路中的仿真操作，观察是否能够达到自己期望的目的，在编写过程中，首先对8连管进行编程，调试好能够正常显示需要的内容，因为之后的调试工作都依赖于数码管的显示，将数码管单独设置为一个模块，进行模块化程序设计，有利于之后对其添加新的功能，并且在出现问题时可以快速找到问题的位置，方便程序的移植以及修改扩展操作。

在完成8连管的程序编写后可以进行按键的编写，在单片机的编程中延时操作是及其浪费单片机的有限资源，但是对于按键来说，因为会有仿真时的一定问题（并且在实际中机械按键会有一定的机械振动，造成在按下按键的时后会检测到多个按的动作的现象，因此需要进行对此问题的处理，即进行消抖操作），一般进行消抖的方式有硬件消抖，即与按键并联一个容值20pf左右的电容进行滤波以达到消抖的目的，但是对于单片机的开发来说这会增加硬件的成本，因不常使用这种硬件滤波的操作，另一种进行消抖的软件方式即进行延时操作，通过延时避免在机械振动过程中检测到错误按键操作，但是对于单片机来说，CPU资源极其珍贵，在延时过程中CPU将会处于一种等待的状态，及其浪费其资源，而且，单片机嵌入式系统重要的便是其实时性的特点，当程序中出现大量的延时操作无疑会对其实时性产生巨大的影响，因此使用了一定时间检测一次之后与之前对比的按键检测方式，这样就会避免延时操作的使用，能够在保持实时性的同时保证按键不会产生错误操作。

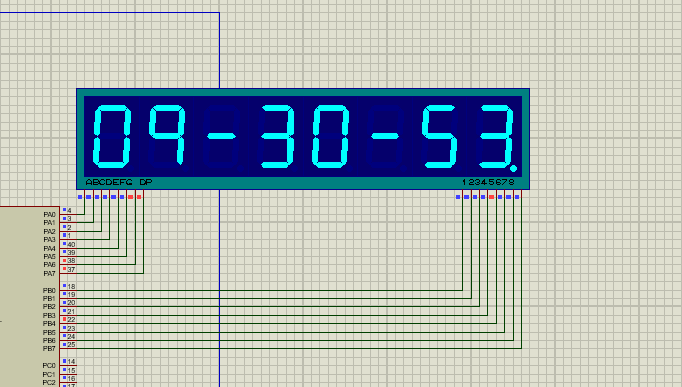
之后就是对蜂鸣器的调试操作，蜂鸣器是有源的因此只需要P17引脚给出对应的电平即可对蜂鸣器是否发声进行控制。至此所有的模块都调试清楚了，之后就是对整体功能的进一步协调，在整体调试的过程中可能会出现一堆的问题，使用keil 5的单步调试功能能够进行单步的跟踪调试，这样在出现问题的时候可以直接观察到自己的问题所在，因此使用调试功能对与整个系统的组合有重要的作用。在调试的过程中最重要的就是需要一定的耐心，因为当你解决掉一个问题后可能又会出现另一个问题，这样的调试是十分枯燥的，但是这就是程序设计的最常见的状态，一般写代码就是比较快但是对与错误与警告的调试将会花费大量的时间，有时候之前的方案不能够很好的适应工程地设计也要善于发散思维从不同的方向对其进行处理，这样在不同的方式下才能够很好的进行程序的设计，编写出符合自己需求的程序。

在调试的过中可以使用keil 5与protues管连起来进行实时的调试操作,这样能够精确的定位到程序问题所在,不仅可以加快程序的准确度,而且可以加快程序的调试过程。在调试过程中会出现由于是软件仿真的问题程序运行与实物有一定差距，而且在仿真时电脑的运行状态等都会影响程序运行后表现出来的效果。因此对于程序的仿真应当多次实验，了解到每一个步骤中应该会有什么现象，做到每一步都心中有数，这样再结合单步调试将会出现事半功倍的效果，让调试中发现单片机开发的乐趣，有利于对单片机内部的认识，提升自己对嵌入式系统的了解情况，不局限与课本上的内容，在调试过程中恰当的使用keil 5自带的调试工具将会十分方便对按键等外部输入的调试工作，直接改寄存器中的对应的值，跳转到指定的位置，在学习调试步骤的过程中对单片机的运行机理进行了进一步的理解，了解到指令执行的第一步，了解到指令的取指，对数据的获取以及计算，对数据的存储，对单片机运行的过程中的堆栈的分布有了进一步的理解，充分认识到中断，寄存器存储原理，PC指针等在单纯学习C语言中难以理解的东西，虽然8051单片机是最基础最简单的8位单片机，但是能够实现的功能却丝毫不少，当多个8051互相配合能够做出出人意料的东西，就像8086的PC机一样，8051单片机是进入嵌入式世界的一个基础，也是重要的一环，了解了8051单片机的运行机理，存储原理，以及运行取指等操作的具体内部内容后对于其他类型的单片机将会十分容易的上手。

电子时钟操作手册

**一：基础部分：**

电子时钟最基础的功能：计时，当程序上电将会自动计时如正常的时钟一样。

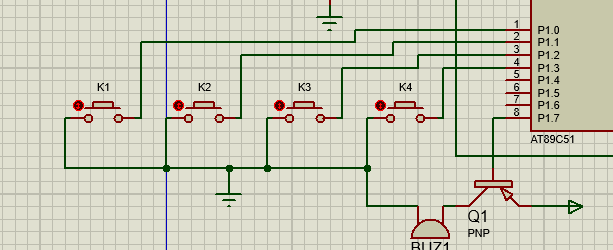
显示内容：小时-分钟-秒

后面的小数点是闹钟开启的标志使用按键K4能够对闹钟是否开启尽行操作，当闹钟关闭的时候小数点不显示。

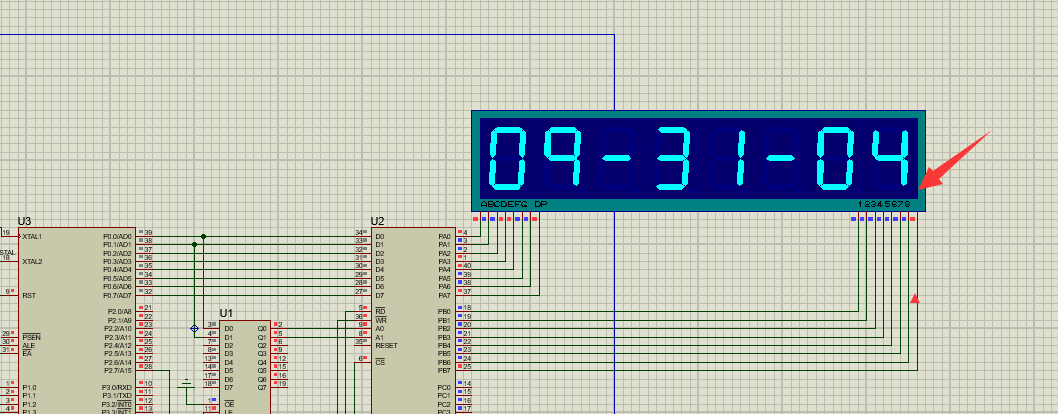
在没有进行按键操作的时候，时钟会按照年份显示3秒日期显示7秒的形式交替显示。

**二：按键**

此电子时钟使用了4个独立按键来进行整体的参数修改配置。按键链接在P10 ~ P13口，使用隔时检测的方式进行检测，保证了按键的功能，并且极大的提升了时钟的整体实时性，增加了时钟的精度。按键排布方式如下

按键上面使用标识标明K1 ~ K4提供人性化操作界面，方便用户的使用。

1. **按键功能描述：**
   1. K1： 按键1用来进行时间与年份的顺时切换操作，使用K1能够在显示年份时切换到时间显示，也能在显示时间时切换到年份，每一次按键按下的10秒内能够随意切换，如果没有按键操作，在10秒后显示将返回按键操作之前显示的内容继续进行显示。
   2. K2： 按键2是用来进行12/24小时显示的切换键，当按下按键时间显示的显示方式将会进行改变。
   3. K3： 按键3是对时间年份的修改操按键，当按下K3键之后K1键将会变成方向选择键，每按下一次K1键将以此选择年，月，日，时，分，秒，进行该数操作。该数的按键是K2，当使用K1键选择要改变的内容后使用K2可以对其进行增加操作，当修改后再按下一次K3键即忽视掉本次修改，可以继续进行修改操作，若按下两次K3将退出数值修改模式，返回之前的显示内容，若在更改之后使用K4键则确定之前的更改，并且直接切换到更改后的进行显示

（注：每次进行修改时都需要对所有位进行修改，不进行修改的位默认为0）

d） K4： 按键4是进行闹钟控制的按键，按下K4将会观察到最后一个数码管的小数点位置将会点亮或者关闭，当点亮时闹钟打开，不亮时闹钟关闭，当闹钟响起按下按键K4将会关闭闹钟，若不进行按键操作30秒后闹钟将会自动停止。