# [滚动字符的广告牌设计报告]

2019年6月20日

专业名称：[电子信科学与技术]



指导教师：[荣生辉]

作者:[鲁国锐，马健康，钱军]

目录

**[一、 摘要](#_Toc21576_WPSOffice_Level1)** **[3](#_Toc21576_WPSOffice_Level1)**

**[二、 任务分析及设计思路](#_Toc8774_WPSOffice_Level1)** **[3](#_Toc8774_WPSOffice_Level1)**

**[三、 硬件设计分析](#_Toc21752_WPSOffice_Level1)** **[3](#_Toc21752_WPSOffice_Level1)**

[1、74HC595](#_Toc8774_WPSOffice_Level2) [3](#_Toc8774_WPSOffice_Level2)

[2、 中断](#_Toc21752_WPSOffice_Level2) [4](#_Toc21752_WPSOffice_Level2)

**[四、 软件设计分析](#_Toc32613_WPSOffice_Level1)** **[4](#_Toc32613_WPSOffice_Level1)**

[1、向74HC595输送数据](#_Toc32613_WPSOffice_Level2) [4](#_Toc32613_WPSOffice_Level2)

[3、 滚动显示](#_Toc22595_WPSOffice_Level2) [4](#_Toc22595_WPSOffice_Level2)

[3、中断函数](#_Toc9773_WPSOffice_Level2) [5](#_Toc9773_WPSOffice_Level2)

[4、软件设计流程图](#_Toc20494_WPSOffice_Level2) [5](#_Toc20494_WPSOffice_Level2)

[图 1 软件设计主函数流程图](#_Toc32766_WPSOffice_Level2) [5](#_Toc32766_WPSOffice_Level2)

**[五、 实验仿真结果及分析](#_Toc22595_WPSOffice_Level1)** **[6](#_Toc22595_WPSOffice_Level1)**

## 摘要

本次作业要求利用Proteus 和 Keil 软件，自行设计一套基于51单片机的滚动字符广告牌。要求其可以利用16×16的LED点阵滚动显示小组成员姓名，滚动显示的方向可以通过外部按键等输入设备进行调整（四种方向）。

## 任务分析及设计思路[[1]](#footnote-0)

我们小组设计的任务是要利用51单片机和4个8×8点阵组成的16×16点阵滚动显示小组成员名字，并且滚动的方向可以通过外部输入设备进行调整。

对于这样一个任务我们首先需要明确点阵显示图形的原理。所谓显示图形，实质上就是把特定位置的LED灯点亮。我们的做法是先选中一行（列），即向对应的管脚输出高电平或低电平，取决于该点阵是行共阳还是行共阴；然后再选中对应的列，同样是向对应的管脚输出相应电平。区别是这样做时一般只会选中一行，但可能选中多列。这样做也是为了显示需要。由于结构本身的限制，我们不可能同时对多行多列进行操作。所以只能把一个图形拆成一行一行来扫描。而由于人眼有视觉暂留效应，只要扫描的速度够快，我们看到就是一副完整的图像。

当然，也可以先选中列再选中行，道理是一样的。这里我们两种方法都要用到。

以上说的是静态显示，而滚动显示就是当扫完最后一行后回到第一行时，对应列需要点亮的是上一轮的第二行的位置。同样，第二行要点亮上一轮第三行的位置，第三行要点亮上一轮第四行的位置……最后一行要点亮上一轮最后一行的下一行需要显示的位置。相当于整个图形向上移动了一格。

这样我们就完成了从上往下的滚动显示，从下往上其实就是刚好把上述过程反过来：第二行点亮上一轮第一行的位置，第三行点亮上一轮第二行的位置……

而左右滚动的原理也是类似的。只不过此时我们是先选中一列，再选中对应的若干行，一列一列进行扫描。相当于把上下滚动中的行列地位交换了。

## 硬件设计分析

## 1、74HC595

此次课程设计中我们用到的芯片有51单片机，4个8×8的LED点阵以及74HC595串行输入、并行输出的位移缓冲器。

51单片机和点阵的作用自不必说。这里需要用到74HC595的原因是一个16×16的点阵需要单片机提供32个接口去分别控制它的16行和16列，但显然51单片机做不到这一点。所以此时我们需要用74HC595来对接口进行扩展。

74HC595可以通过DS引脚接受单片机输入的串行数据，然后把得到的8位串行数据转成并行数据再输出。这样单片机只需要提供一个引脚就可以输出8位数据，相当于把一个引脚当八个来用。

同时这里我们对74HC595进行了级联，两个一组能够同时输出16位数据。这样我们就可以只用两个单片机接口来分别控制16×16点阵的行和列了。

当然，这里对每一组74HC595还需要两个时钟信号来对DS引脚上的信号进行接收和移位处理。所以总共要用到六个接口。

## 中断

51单片机的外部中断电路比较简单，只需要通过按钮把P3.2和P3.3引脚接地即可。

## 软件设计分析

### 1、向74HC595输送数据

这里我们为了简化程序，需要专门写函数来给74HC595输送数据。由于上下滚动和左右滚动在输送数据时稍有不同，所以我们总共写了四个函数，分别用于在上下滚动时向行列送数据和在左右滚动时向行列输送数据。

以上下滚动为例，首先我们需要选中一行。这时需要往控制行的那一组74HC595输送一个16位的数，被选中的哪一行对应位是0，其余为1。由于要一位一位地输出，所以我们每次先把要输送的数跟8000H相与，即只保留最高位，把计算的结果输出，然后令待输出的数左移移位，再重复上述过程。

需要注意的一点是，在每次输出的过程中，要令控制74HC595移位寄存器时钟和存储寄存器时钟的变量置0、置1，否则数据无法被正确读入。

向控制列的那一组74HC595输送数据的函数基本与上面所说的相同。只是此时向列输送的是字码，而字码我们在计算时是以8位二进制数的形式存放的。所以在输送前要与80H相与。

同时在调用这个函数时也要注意，输出一行的字码要用两次该函数。

左右滚动与上下滚动输出数据的过程基本一致，只不过此时向控制行的那组74HC595输出的是字码，所以在该函数中待输出数据应与80H相与；而相应的，向控制列的那组74HC595输出时要与8000H相与。

另外在左右滚动时还有一点需要特别注意，选通行时要输出低电平，选通列时要输出高电平，而我们的字码和选通行列的数据都是按上下滚动时计算的，即字码是为列选通而计算的，用于选通的数是按行来计算的。前面我们说过，左右滚动时行列的地位要颠倒，所以此时对字码和用于选通的数都要先按位取反再作为函数参数调用。

## 滚动显示

点阵要进行滚动显示就意味着，每一轮向点阵输出的16行字码要整体向后挪一位。所以我们这里定义一个变量，每一轮以这个变量的值为起始，向后输出16行字码。然后再令该变量加2（一行字码用的是两个8位的二进制数存放的，故应加2）。同时为了保证输出完所有字之后能循环回第一个字，当该变量的值要超过字码数组的值时我们把它置0。而如果要反向滚动，则每次令该变量减2即可。相应的，为了使反向的滚动也能够循环，我们在该变量小于0时把它置为数组的长度减一（原因仍然是两个字码为一行）。

这里有一个特别容易被忽略的地方，就是原作者在定义这个变量时用的是unsigned int类型，所以它在减到0后不会出现负数的情况，而是会溢出。所以这里我们不能直接判断它是否为负数。但是仔细考虑下我们会发现，溢出后的值肯定是大于数组长度的，所以在正向滚动时用的那个判断条件这里仍可以用，只是把置0改为置数组长度减一。

## 3、中断函数

电路设计中我们用到了两个中断源，分别控制上下滚动的方向和左右滚动的方向。所以这里我们也要用两个变量hang\_flag和lie\_flag来表示上下滚动和左右滚动的方向。这里我们用1来表示从上往下或从左往右，用-1表示从下往上或从右往左。所以在中断函数中我们要做的其实就是令相应的标志乘上-1即可。同时还要注意的是，在调用控制行方向的中断时，要令lie\_flag为0；在调用控制列方向的中断时，要令hang\_flag为0。

最后要强调的一点是，外部中断1对应的中断号是2。之前想当然地以为是1，结果左右滚动怎么都出不来，在这上面卡了很久。

### 4、软件设计流程图

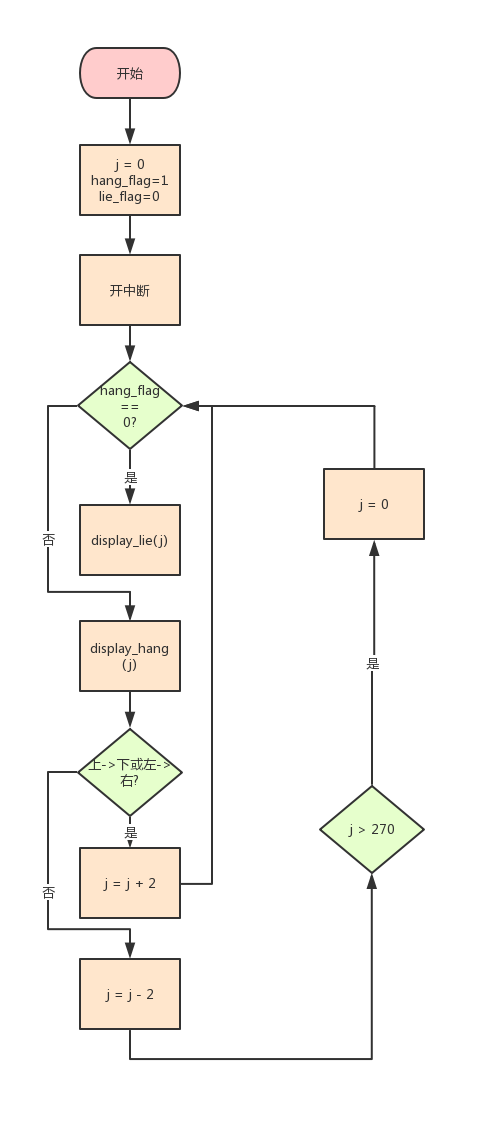


图 1 软件设计主函数流程图

## 实验仿真结果及分析

实验结果跟预期一致，原件基本能够正常运行，也能通过按钮来控制滚动的方向。但还存在着一个问题，就是在滚动的过程中字的显示会不太清楚，会有一些本不应该在此刻点亮LED灯被点亮。而在我们单独实现每个方向时似乎没有这样的情况。具体原因目前还不太清楚。

1. 我们的任务设计是在别人设计的基础上进行改进得来的。原设计的网址：<https://blog.csdn.net/qq_41639829/article/details/82726377?tdsourcetag=s_pcqq_aiomsg> [↑](#footnote-ref-0)