16*16点阵LED的设计

张 琛 耿 标
（中国矿业大学 信息与电气工程学院 江苏 徐州 221000）

摘 要： 设计一个16*16的LED点阵作为显示屏的最小模块，控制器采用AVR单片机，还有相应的驱动芯片74LS154和74LS595分别作为点阵LED的行选和列选，设计电路简单可靠，可扩展，级联，能够作为LED显示屏设计的最小模块。

关键词： 16*16LED；74LS154；74LS595；AVR单片机；8550三极管
中图分类号： TN8 **文献标识码：** A **文章编号：** 1671－7597（2010）0810064－01

0 引言

无论实在日常生活中还是在工业现场，LED屏处处可见，通常LED显示屏是由显示模块、控制系统及电源系统组成。显示模块由LED灯组成的点阵构成，负责发光显示；控制系统通过控制相应区域的亮灭，可以让屏幕显示文字、图片、视频等内容，恒舞动卡主要是播放动画的；电源系统负责将输入电压电流转为显示屏需要的电压电流。由于LED显示屏幕的亮度高、成本低、可靠性好、使用方便等特点在市场上毫不逊色与LCD液晶显示，甚至可以说，在某些特定的场合只能选用LED作为显示屏幕，这和其很强的稳定、可靠性息息相关。

所以，LED显示屏幕的驱动设计很有意义。本文主要讨论如何设计一个16*16的点阵LED模块，实际上，显示屏幕也是由若干个点阵模块拼接而成，因此，取其典型，我们只讨论一个最小模块的设计。

1 系统硬件设计

点阵LED的设计核心问题在于驱动电路的设计，驱动电路设计的成功与否直接影响到LED的显示效果。LED的显示效果主要讨论的是其亮度和色彩的明亮度，而亮度主要取决于驱动电流的大小，所以提供合理的驱动电流尤为重要。

这里我们可以估算一下，一个LED点亮所需的驱动电流大约为10-20mA，这里采用的是16*16的点阵，由于各列LED并联，所以行驱动至少要提供16*10=160mA的电流；由于同列LED串联，所以列驱动只需要提供本列。的驱动电流即可，即大约10mA。虽然AVR单片机的驱动能力很强大约为20mA左右，但对于庞大的点阵来说是远远不够的，所以我们最常用的方法就是选用三极管扩流，这里可以选用8550。

系统硬件框图如图1。

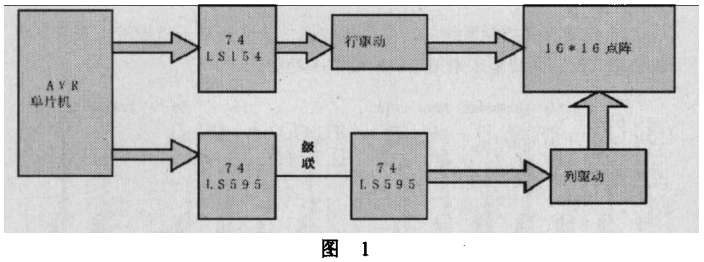

图 1

硬件说明：74LS154为4/16译码器，低电平译中，作为16行的行选通信号；74LS595为8位可级联的串入并出的移位寄存器，所以在本实验中需要两片74LS595级联，作为16列的选通信号，通过AVR单片机来控制74LS595移位寄存器和74LS154译码器即可完成16*16点阵LED的驱动。

Proteus仿真图：如图2。

2 软件设计

16*16的点阵LED关键部分是硬件电路的设计，软件设计就相对要灵活的多，硬件搭建起后，我们就可以对AVR单片机编写相应的程序来实现不同的显示功能，还可以添加很多的动态效果，比如卷入卷出、滚动，帘出帘入等等，只要程序可以办的到的它都能实现。

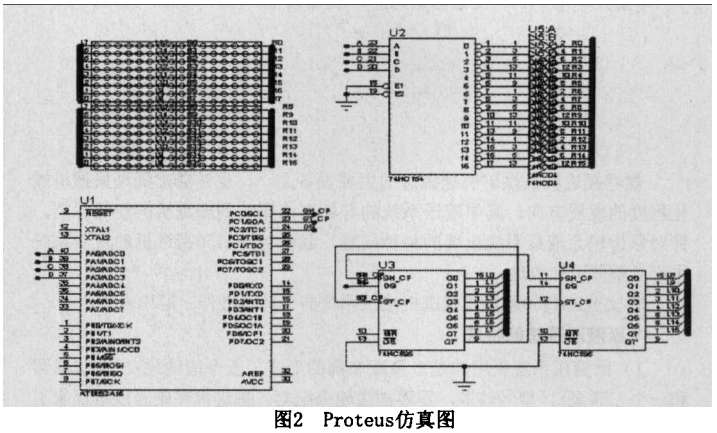

图2 Proteus仿真图

本文主要介绍软件驱动程序的一个关键部分，就是74LS595的移位操作，我们是通过74LS154的译码输出来选通行，相对应我们要对74LS595写一个字，使其选通相应的列，这样就完成了一次LED的扫描操作，由于人的视觉暂留，只要设置不同的扫描周期就可以产生不同的动态效果。

扫描周期的确定可以是模糊的，也可以是精确的。所谓模糊的，就是根据经验，和人的视觉暂留的时间来估计延时，可能需要经过几次的尝试才能达到较好的显示效果。所谓精确的就是根据人的视觉暂留时间，和所扫描的点阵数量来精确计算扫描周期，通过定时器来实现较为精确的扫描周期。这种方法耗时一般不用这么精确，选择模糊的判断和几次尝试就能达到很好的显示效果了。

74LS595的移位操作程序：

```
Void Shift_595(uchar*p)//p为查表指针
{unsigned chardata_h, data_l; unsign ed int i; data_h=*p; data_l=*(p+1);
PORTC&=~BIT(SH_CP); //移位时钟拉低
PORTC&=~BIT(ST_CP); //锁存时钟拉低
for(i=0; i<8; i++)//先移高位
{if(data_h&0x80)//如果最高位为1，则移出1，即DS=1
PORTC|=BIT(DS);
Else
PORTC&=~BIT(DS); //否则为0，即DS=0
PORTC|=BIT(SH_CP); //上升沿移位
PORTC&=~BIT(SH_CP);
data_h=data_h<<1;              //数据左移一位，循环
}
for(i=0; i<8; i++)                    //再移低位
{if(data_l&0x80)              //如果最高位为1，则移出1，即DS=1
```

（下转第59页）

件可支持的最低性能的硬件，等等。

3）安装测试用例

需要核实测试目标可以在所有可能的安装情况下安装/反安装：安装情况可以指首次安装测试目标，或是在装有较早版本的机器上安装测试目标的某个较新的版本或工作版本；安装测试还应确保在遇到异常情况时（如磁盘空间不足），测试目标的执行情况仍可接受；以及上述条件下的反安装测试。

4）安全测试用例

安全测试，在当前各种复杂的应用、网络环境下，其重要性越来越显著。相应的软件是否能够按照预期完成软件的权限控制，在存在攻击等“非正常”环境下是否仍能够保证软件的安全性。

2.2.2 性能测试用例的设计过程

1）对于软件指标中明确说明的性能标准的各条说明都应确保至少要确定一个测试用例，这是最基本的测试用例设计要求。

2）对每个关键应用，都应确保至少要确定一个测试用例。

常对于每个用例/需求都会存在不止一个性能测试用例。前面也提到，除了软件指标中明确要求的性能点之外，在高负载、多攻击等压力环境下，系统（软件）是否仍能保持应有的性能，是我们在设计测试用例时更需要关注的，也是通常软件表现容易出问题的地方。

2.2.3 其它测试用例的生成

除了功能、性能测试用例之外，还有可靠性测试用例。同时，对于有经验的测试用例设计人员来说，仍可能找到其他需要补充的测试用例，如：在发生故障后软件的表现、对性能瓶颈或超出软件极限情况进行调查的测试用例。

2.3 测试用例设计的检查点与注意点

测试用例需要经过一定的检查与评审过程才能交付使用，测试用例检查点即用于评估测试用例质量的一些关键内容，主要包括：

1）对于每个测试需求，是否都已覆盖；同时至少需要确定正面和负面各1个测试用例，或者依据软件的实际情况定义不同的可量化的要求。

2）测试用例的各种测试类型是否都已设计，包括：功能、功能交互、数据流、对象状态、性能、安全性、压力、稳定性、兼容性等几个方面。

3）每个测试用例的测试方法、测试结果是否正确，能否达成测试目的。

4）每个测试用例是否清晰描述了测试用例测试的内容，以及被测对象预期的行为。

5）每个测试用例是否定义单一的输入集合，或者操作，并得到唯一的测试结果，用于检查测试用例是否具备执行的可重复性。

6）每一个测试用例的测试结果的验收方法是否明确？验收手段是否全面？

7）所有的测试用例名称和/或ID都与测试工件命名约定一致。

除了测试用例审查的检查点外，在测试用例设计过程中，还存在着一些需要遵循的常用原则：

1）通常应该避免测试用例的相互依赖；

2）测试设计应注意的是输入为测试需求（可包括各类指标、设计文档等）。在某些情况下（如测试用例的实施验证）可能需要将试验软件（如仍处于集成测试阶段的软件）作为测试设计过程的输入，但测试设计人员必须注意并不是在测试实际的软件实现，软件实现结果如何只能证明其完成的工作，并不是其应该完成的工作。

3 如何描述测试用例

本章节说明对于每组（个）测试用例应如何描述。简单地说，从测试用例设计的定义、目的与包含的内容出发，每个测试用例的组成应包括如下3个基本元素：

1）测试用例说明——说明这组数据实施/执行的步骤、目标与预期结果；

2）测试输入——在执行该测试用例时，输入的与之交互的对象、字段和特定数据值（或生成的对象状态）；

3）预期结果——执行该测试用例完毕后得到的状态或数据。

对上述内容进一步细分，包含：测试用例名称、测试用例描述、测试说明、环境需求（测试模型或测试拓扑）、前置条件、输入数据（测试数据可以体现在测试步骤中，若有预先已存在的数据，如：配置文件等，可在测试用例中予以说明引用的测试数据文件）、测试步骤与预期结果。

同时，出于测试用例维护与定义的目的还需要包括用例标题与标识符（必须唯一）、版本修订历史记录（版本号、修订时间、修改记录与作者等）。

另外，出于整体测试用例阅读与执行的便利性考虑，还可包括：用例等级、测试用时、用例间的相互关系等内容。

综合以上内容，可以设计适用于实际的测试用例模板，具体的格式可以依据实际情况确定，内容可参考以上所列出的信息。从笔者的软件测试实践中看，简单地说，测试用例的组织与说明方式，不外乎围绕着其可读性、易用性角度出发来进行，同时兼具测试用例的可管理性等内容，考虑了这几个方面，测试用例的组织与说明就能够比较清晰明了。

本文对测试用例组织、设计方式方法的说明，来源于笔者对软件测试的一些基础理论与软件测试过程实践的结合，希望能够给软件产品测试用例设计提供一定指导与帮助。

参考文献：

[1]IEEE Std 610.12-1990（IEEE Standard for Glossary of Software Engineering Terminology）.

[2]IEEE Std 829-1998（IEEE Standard for Software Test Documentation）.

[3]郑人杰，《计算机软件测试技术》清华大学出版社，1992年.

[4]RexBlack（龚波、但静培、林生、周志全等译）《软件测试过程管理》机械工业出版社，2003年.

作者简介：

张晓敏（1978-），女，汉族，福建福州人，福建星网锐捷网络有限公司，工程师。

（上接第64页）

```
PORTC|=BIT(DS);
else
PORTC&=~BIT(DS);                    //否则为0，即DS=0
PORTC|=BIT(SH_CP);              //上升沿移位
PORTC&=~BIT(SH_CP);
data_l=data_l<<1;                   //数据左移一位，循环
}
PORTC|=BIT(ST_CP);      //两个字节数据锁存数据、输出
PORTC&=~BIT(ST_CP);
}
```

3 总结

本实验通过Proteus仿真，实现了简单的16*16LED控制，由于仿真不能模拟电流的大小对显示效果的影响，所以通过实物制作，获得了很好的显示效果，这种方案是可行的，而且简单，便于扩展，对了解LED显示屏的工作原理和结构很有帮助。

参考文献：

[1]楼然苗、李光飞，单片机课程设计指导，北京航空航天大学出版社，2008.

[2]张军，AVR单片机应用系统开发典型实例，中国电力出版社，2005.