基于51单片机的8×8 LED显示

杨题鸣（201883016）

（大连理工大学 电信信息工程与电气工程学部 电子信息工程专业，辽宁 大连 116024）

**摘 要**：用51单片机的P0口外接74LS245作为控制8×8 LED点阵屏的行驱动,列选通由P1端口控制,程序运行时,8×8 LED点阵屏依次循环显示‘大’‘工’‘N’‘O’‘1’字样。

**关键词：**8×8 LED，51单片机，c语言，keil5，proteus

**8×8 LED display based on 51 MCU**

**Abstract:** I The P0 port of 51 MCU is connected to 74LS245 as the row driver controlling the 8×8 LED lattice screen. The column selection is controlled by port P1. When the program is running, the words "big", "work", "N", "O" and "1" are displayed in turn on the 8×8 LED lattice screen.

Key words: 8×8 LED, 51 MCU, C language，keil5，proteus

1 总体设计方案与要求

1.1设计课题任务

用51单片机的P0口外接74LS245作为控制8×8 LED点阵屏的行驱动,并显示8×8 LED点阵屏依次循环显示‘大’‘工’‘N’‘O’‘1’字样。

1.2设计课题总体方案思路系统框图以及总体方案说明

1.2.1方案思路

8×8LED点阵屏显示数字系统，可用单片机直接控制信号灯的状态变化可以广泛的应用到商业和工业的流程控制测电路当中。

原理框图：

双向总线收发器

时钟

输入

复位

信号

AT89C51

列码扫描

行码数据

8×8LED点阵

双向总线收发器

图 系统的总体框图

2 设计电路及其仿真

2.1 设计总电路图

图示, 示意图

描述已自动生成

基于51单片机的8×8 LED电路图

2.2重要模块介绍：

2.2.1单片机时钟与复位电路部分

图示, 示意图

描述已自动生成

时钟电路功能：

只要单片机电路接通电源，时钟电路开始振荡工作，就为单片机提供源源不断的时钟脉冲。

单片机内部都是由许多触发器等构成的时序电路组成的，只有通过时钟才能使单片机一步步地工作。单片机时钟信号好比是单片机的心脏,单片机时钟频率决定了单片机运行一个指令周期所需的时间。

复位电路功能：

电路通电瞬间对电解电容充电，由于电解电容上的电压不能突变，相当于瞬间短路，也就是5V高电平通到单片机复位脚 上，电容在慢慢充电过程（RC乘积时间），复位脚上的电压会慢慢降低。只要复位脚上的高电平时间保持在2微秒（2us)就能可靠地复位（外部时钟为12MHZ时）

2.2.2 AT89C51芯片部分：

图表

描述已自动生成

89C51是一种带4K字节闪存可编程可擦除只读存储器（FPEROM—Flash Programmable and Erasable Read Only Memory）的低电压、高性能CMOS 8位微处理器，俗称单片机。AT89C2051是一种带2K字节闪存可编程可擦除只读存储器的单片机。单片机的可擦除只读存储器可以反复擦除1000次。该器件采用ATMEL高密度非易失存储器制造技术制造，与工业标准的MCS-51指令集和输出管脚相兼容。由于将多功能8位CPU和闪烁存储器组合在单个芯片中，ATMEL的AT89C51是一种高效微控制器，AT89C2051是它的一种精简版本。AT89C单片机为很多嵌入式控制系统提供了一种灵活性高且价廉的方案。

AT89C51单片机为40引脚双列直插芯片，有4个I/O口，P0 ，P1， P2， P3，单片机的最小系统如图所示，18引脚和19引脚接时钟脉冲电路，XTAL1接外部晶振和微调电容的一端，在片内它是震荡器倒相放大器的输入，XTAL2接外部晶振和微调电容的另一端，在片内她是振荡器倒相放大器的输出端，第９引脚为复位输入端，接上电容，电阻及开关后构成上电复位电路，20引脚为接地端，40引脚为电源端。

2.2.3 51单片机输出部分

图示, 示意图

描述已自动生成

**74LS245：**

74LS245是用来驱动LED或者其du他的设备，它是zhi8路同相三态双向总线收发器，可双向传输数据。74LS245还具有双向三态功能，既可以输出，也可以输入数据。 当8051单片机的P0口总线负载达到或超过P0最大负载能力时，必须接入74LS245等总线驱动器。P0口与74LS245输入端相连,E端接地，保证数据线畅通。

**RESPACK-8**

RESPACK-8是一种排阻。排阻用于数字电路，集成若干单一电阻，内部方式可以串联，或者并联；简化PCB板设计、安装更加方便、保证SMT 焊接质量、减小成套设备的体积。阻抗匹配后对本级信号基本无影响。

**8x8LED:**

主要用于显示LED

根据本设计的特点，LED点阵的显示不可少，LED的点阵显示采用普通的发光二极管。在硬件上连接图上也是对称分布的，如下图3.3所示。

图片包含 背景图案

描述已自动生成

2.4 keil5代码编写

代码如下：

#include <AT89X52.H>

**unsigned** **char** code taba[]={0xfe,0xfd,0xfb,0xf7,0xef,0xdf,0xbf,0x7f};//选行

**unsigned** **char** code tabb[]={0x18,0x18,0xFF,0xFF,0x18,0x3C,0x66,0xC3};//字体’大’

**unsigned** **char** code tabc[]={0xFF,0xFF,0x18,0x18,0x18,0x18,0xFF,0xFF};//字体’工’

**unsigned** **char** code tabd[]={0xC3,0xC7,0xCF,0xDF,0xFB,0xF3,0xE3,0xC3};//字体’N’

**unsigned** **char** code tabe[]={0x3C,0x7E,0xC3,0xC3,0xC3,0xC3,0x7E,0x3C};//字体’O’

**unsigned** **char** code tabf[]={0x1C,0x1E,0x1F,0x19,0x18,0x18,0xFF,0xFF};//字体’1’

**unsigned** **char** i,j;

**void** delay(**void**)//延迟1

{

**unsigned** **char** i,j;

**for**(i=10;i>0;i--)

**for**(j=24;j>0;j--);

}

**void** delay1(**void**) //延迟2

{

**unsigned** **char** i,j,k;

**for**(k=10;k>0;k--)

**for**(i=20;i>0;i--)

**for**(j=24;j>0;j--);

}

**void** main(**void**)

{

**while**(1)

{

**for**(j=0; j<10; j++) //行扫描

{

**for**(i=0;i<8;i++)//字体‘大’扫描

{

P2=taba[i];

P3=tabb[i];

//P1=0xff;

delay();

}

}

**for**(i=0;i<8;i++)

{

P2=taba[i];

P3=0xff;

delay1();

}

**for**(j=0; j<10; j++) //行扫描

{

**for**(i=0;i<8;i++)//字体‘工’扫描

{

P2=taba[i];

P3=tabc[i];

//P1=0xff;

delay();

}

}

**for**(i=0;i<8;i++)

{

delay1();

}

**for**(j=0; j<10; j++) //行扫描

{

**for**(i=0;i<8;i++)//字体‘N’扫描

{

P2=taba[i];

P3=tabd[i];

//P1=0xff;

delay();

}

}

**for**(i=0;i<8;i++)

{

P2=taba[7-i];

//P3=tabb[i];

P3=0xff;

delay1();

}

**for**(i=0;i<8;i++)

{

delay1();

}

**for**(j=0; j<10; j++) //行扫描

{

**for**(i=0;i<8;i++)//字体‘O’扫描

{

P2=taba[i];

P3=tabe[i];

//P1=0xff;

delay();

}

}

**for**(i=0;i<8;i++)

{

delay1();

}

**for**(j=0; j<10; j++) //行扫描

{

**for**(i=0;i<8;i++)//字体‘1’扫描

{

P2=taba[i];

P3=tabf[i];

//P1=0xff;

delay();

}

}

**for**(i=0;i<8;i++)

{

delay1();

}

}

}

2.4 设计电路图的仿真结果

如图所示：最终结果下图所示显示“大工NO 1”

图片包含 游戏机

描述已自动生成

3 结语

课设总结与感悟

本次电子系统方针，接触proteus 是新的篇章的开始，第一次使用proteus ，发现了新大陆，比起multisim，proteus的使用上更加友好，而且还能编程，打开了我新的单片机学习生涯。

通过单片机课程设计，我不仅加深了对单片机理论的理解，将理论很好地应用到实际当中去，而且我还学会了如何去培养我们的创新精神，从而不断地战胜自己，超越自己。创新，是要我们学会将理论很好地联系实际，并不断地去开动自己的大脑，从为人类造福的意愿出发，做自己力所能及的，别人却没想到的事。使之不断地战胜别人，超越前人。同时，更重要的是，我在这一设计过程中，学会了坚持不懈，不轻易言弃。设计过程，也好比是我们人类成长的历程，常有一些不如意，也许这就是在对我们提出了挑战，勇敢过，也战胜了，胜利的钟声也就一定会为我们而敲响。

**参考文献(References)：**

1. 51单片机工程创建与Proteus流水灯仿真<https://www.bilibili.com/video/BV12E411i7Ls>
2. 【51单片机100例】－项目36:8×8点阵屏显示数字<https://www.bilibili.com/video/BV1AK411M7qQ?from=search&seid=15803836002083274881>
3. 51单片机8\*8点阵显示（汉字）<https://www.bilibili.com/video/BV1cp4y197Du?from=search&seid=11553419322595319326>
4. Keil5软件安装包以及安装教程https://baijiahao.baidu.com/s?id=1596425032383473114&wfr=spider&for=pc

**基于51单片机的8×8 LED显示纠错**

1.proteus以及keil5安装问题

在proteus安装过程，对于一个重来未接触过proteus的人来说，是一个比较难的事情。安装过程中，遇到的最大的问题，就是找不到文件库，一开始以为是proteus没有自带文件库，所以就在网上下载了文件库，但是发现依旧出现问题，后来发现是因为第一配置的时候，里面所有文件夹指定的位置都是错的，都比实际的文件夹多了一个/data/，运行依旧发现找不到文件夹，然后又更改了管理员权限，终于把proteus正常运行了出来。

keil5是51单片机，编写代码生成hex的一个重要的软件，这个keil5软件的安装主要出现问题在破解上问题上，还有就是如何正确生成一个适用于51单片机的工程文件。学习的资源很少，最终还是在自己的摸索，和在哔哩哔哩网站上面的学习，完成了keil5的安装以及使用。

2.51单片机理解

第一次接触单片机，对他充满了未知。一开始接触以为是和8086一样，采用汇编语言来做。经过漫长的摸索，也找到了相关的学习视频，最终发现实际接触上来，其实发现他比起8086来说，真的好理解得多。51单片机是在c语言的基础上做的。对于一个学过c语言的人来说，51单片机的理解就变得比较简单。

3. 8\*8LED矩阵如何生成问题

为了打出‘大’‘工’‘N’‘O’‘1’的字样，进行了思考，在看过8\*8LED矩阵的原理图后，发现其实对他的理解，就是当成16进制。以每行为例子：每一行有8个灯，那么每一个灯代表一位bit。8个灯就是0FFH，只要在8\*8LED矩阵上面写上该有的字，然后每一行转换成16进制即可。

4.代码理解问题

在网络上，有各种各样的格式基于51单片机的c语言代码。每一个代码，都有自己的思路，一开始很难理解每一种代码的含义。后面还是在blibli上面找到了新手教程，最终选择了一个好理解的代码形式，最终完成编写。