**实验报告三 LED点阵显示屏**

|  |  |  |  |
| --- | --- | --- | --- |
| 姓名 | 学号 | 组内分工 | 工作量占百分比 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

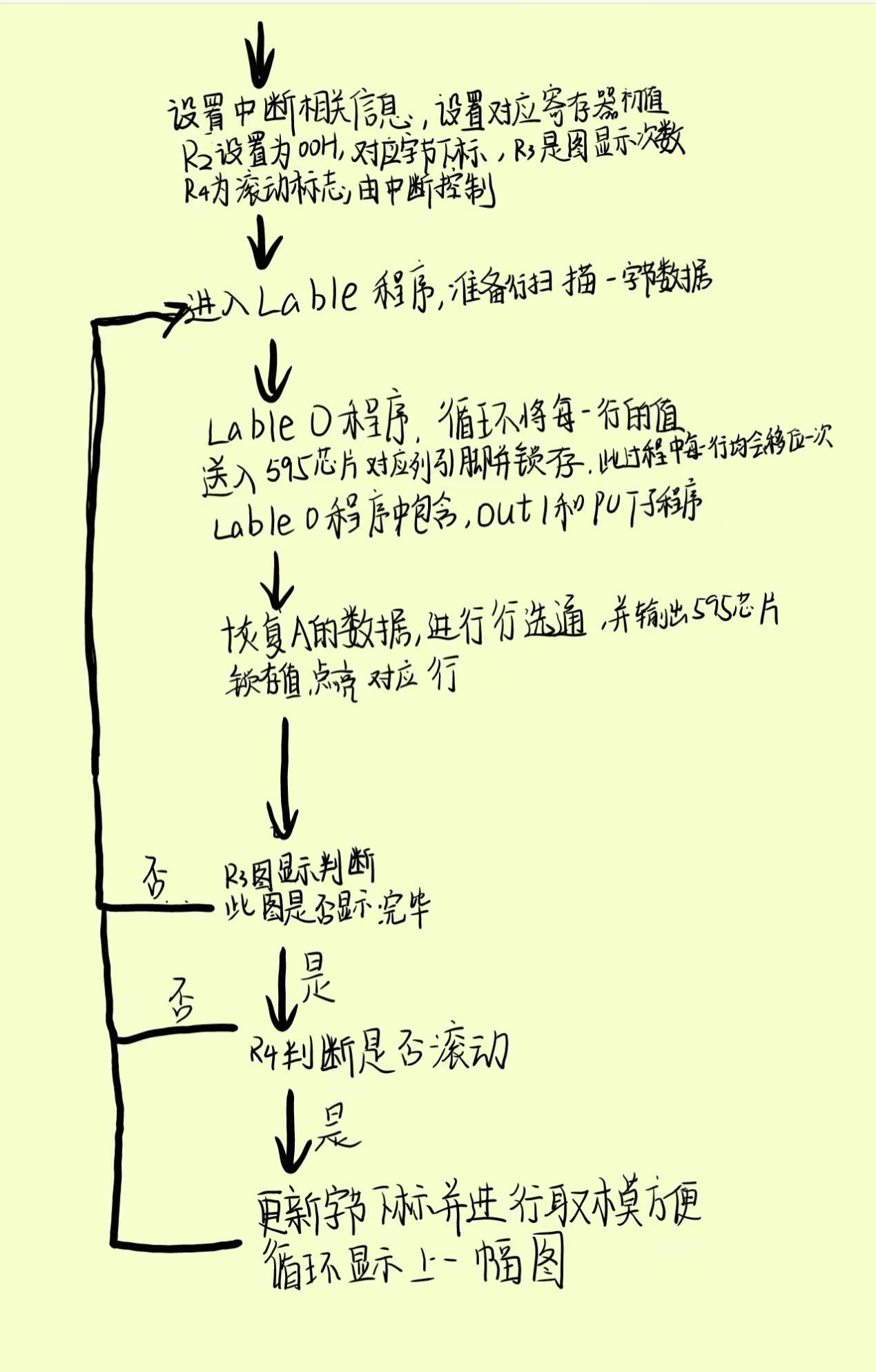
一、电路图

按照要求设计的电路图，及简要介绍。



二、程序分析

1、流程图的形式给出程序设计的思路



2、代码及必要注释。

;====================================================================

; 定义

;====================================================================

SHCP EQU P1.1 ; 移位输入时钟

STCP EQU P1.2 ; 写入时钟

DATAS EQU P1.3 ; 数据线

;====================================================================

; 变量

;====================================================================

;====================================================================

; 复位和中断向量

;====================================================================

; 复位向量

org 0000h

jmp Start

;====================================================================

; 代码段

;====================================================================

org 0100h

; 符号数据

; 每个字表示为16\*16的点阵（亮/暗），一共256bit = 32 Byte

; 而一共有16行，因此每行其实就是 2 Byte

CHARACTER:

DB 004H,002H,0FCH,07EH,012H,009H,0A1H,010H,0FCH,01FH,080H,010H,080H,010H,0FCH,01FH;

DB 084H,000H,084H,000H,0FCH,03FH,0C0H,020H,0A0H,020H,098H,014H,087H,008H,080H,000H;"第",0

DB 080H,000H,080H,000H,080H,000H,080H,000H,080H,000H,080H,000H,0FFH,07FH,080H,000H;

DB 080H,000H,080H,000H,080H,000H,080H,000H,080H,000H,080H,000H,080H,000H,080H,000H;"十",1

DB 040H,000H,080H,000H,000H,001H,000H,001H,000H,000H,0FFH,07FH,000H,000H,000H,000H;

DB 020H,002H,020H,004H,010H,008H,010H,010H,008H,010H,004H,020H,002H,020H,000H,000H;"六",2

DB 008H,000H,088H,01FH,084H,010H,084H,010H,092H,010H,09FH,01FH,088H,010H,084H,010H;

DB 082H,010H,09FH,01FH,082H,010H,080H,010H,098H,010H,087H,010H,0E2H,07FH,000H,000H;"组",3

Start:

; 初始化代码

MOV DPTR, #CHARACTER ; 将数据指针DPTR设置为CHARACTER的地址

MOV R0,#0 ; 初始化行值

MOV A,#0 ; 初始化累加器A

MAIN:

; 主程序循环

PUSH ACC ; 保存累加器A的当前值

CALL WRITE\_BYTE ; 调用WRITE\_BYTE子程序

POP ACC ; 恢复累加器A的值

INC A ; 增加A的值

INC A

CJNE A,#128,MAIN ; 如果A不等于128则继续循环

MOV A,#0 ; 如果A等于128则重置A为0

jmp MAIN ; 无条件跳转到MAIN

; 写字节子程序

WRITE\_BYTE:

MOV R6,#16 ; 初始化行计数器R6为16

MOV R0,#0 ; 初始列地址为0

ROW:

COL:

MOV R7,#8 ; 初始化列计数器R7为8

PUSH ACC ; 保存累加器A的当前值

MOVC A,@A+DPTR ; 从内存指针DPTR指定的位置取数据

; 处理列的第一个字节

COL1:

RLC A ; 循环移位A中的位

;将 A 和进位标志一起向左循环移位 1 位。位 7 移入 CY,CY 移入位 0。不影 响标志

CLR SHCP ; 清除移位输入时钟

MOV DATAS, C ; 将累加器C的值送入数据线

SETB SHCP ; 设置移位输入时钟

DJNZ R7, COL1 ; 如果R7不等于0则继续列处理循环

MOV R7,#8 ; 重新初始化列计数器R7为8

POP ACC ; 恢复累加器A的值

; 增加A的值，获得第二个字节

INC A

PUSH ACC ; 保存累加器A的当前值

MOVC A,@A+DPTR ; 从内存指针DPTR指定的位置取数据

; 处理列的第二个字节

COL2:

RLC A ; 循环移位A中的位

CLR SHCP ; 清除移位输入时钟

MOV DATAS, C ; 将累加器C的值送入数据线

SETB SHCP ; 设置移位输入时钟

DJNZ R7, COL2 ; 如果R7不等于0则继续列处理循环

;STCP上升沿将移位寄存器的数据锁存到输出寄存器

CLR STCP ; 清除写入时钟

NOP ; 空操作

SETB STCP ; 设置写入时钟

POP ACC ; 恢复累加器A的值

INC A ; 增加A的值

CJNE A,#128,HuanHang ; 如果A不等于128则跳转到HuanHang

MOV A,#0 ; 如果A等于128则重置A为0，从头开始显示

HuanHang:

; 切换到下一行，并调用延迟子程序

CALL DELAY ; 调用延迟子程序

;P2连接两块74LS138，用于控制输出的行

PUSH ACC ; 保存累加器A的当前值

MOV A,R0 ; 将当前行号赋值给P0端口

MOV P0,A ; 输出行地址

POP ACC ; 恢复累加器A的值

INC R0 ; 增加R0（行计数器）的值

DJNZ R6, ROW ; 如果R6不等于0则继续行处理循环

RET ; 返回上一级调用

; 延迟子程序

DELAY:

MOV R4, #140 ; 延迟时间计数器初始化

DELAY\_1:

MOV R5, #10 ; 内层延迟计数器初始化

DJNZ R5, $ ; 一直循环直到R5达到0

DJNZ R4, DELAY\_1 ; 循环内层延迟，直到R4达到0

RET ; 返回上一级调用

;====================================================================

END

1. 思考题（见PPT）

1.设计一个128x32的点阵显示屏。

（1）硬件连接：使用74LS595作为列控制芯片，将其串联以扩展输出引脚的数量。连接单片机的GPIO引脚到74LS595的串行数据输入（DS）、时钟输入（SHCP）和锁存输入（STCP）引脚。使用74LS138作为行控制芯片，连接单片机的GPIO引脚到74LS138的地址输入引脚（A0、A1、A2）和使能输入引脚（G1、G2A、G2B）。

（2）引脚分配：将74LS595的输出引脚（Q0-Q7）连接到点阵显示屏的列引脚，连接128个列。将74LS138的输出引脚（Y0-Y5）连接到点阵显示屏的行引脚，连接32个行。

（3）数据传输：使用单片机通过74LS595向点阵显示屏发送列数据。将要显示的图像数据按照列的顺序传输到74LS595的串行数据输入（DS）引脚，并通过时钟输入（SHCP）引脚进行时钟脉冲的控制。然后通过锁存输入（STCP）引脚将数据输出到74LS595的输出引脚（Q0-Q7）。使用单片机通过74LS138向点阵显示屏发送行控制信号。根据要显示的行号，将对应的地址信号（A0、A1、A2）输出到74LS138的地址输入引脚，同时控制使能输入引脚（G1、G2A、G2B）来选择要显示的行。

2.实现一串文字的水平移动：

（1）设定显示区域：确定显示屏上可以显示的最大字符数。假设你的显示屏宽度可以容纳16个字符。为了实现水平移动效果，你需要在显示区域内循环显示一串字符，并通过不断更新字符的位置来实现移动效果。

（2）编写程序逻辑：在单片机上编写程序逻辑，控制字符的显示和移动。首先，初始化显示屏，设置初始的字符位置和待显示的字符串。然后，进入循环，在每个循环迭代中进行以下操作：将当前要显示的字符从字符串中取出。根据字符的点阵数据，在显示区域内绘制该字符。延时一定时间，以控制字符的滚动速度。清除上一个字符的显示，准备下一个字符的显示。

（3）实现字符滚动：不断更新字符的位置来实现滚动效果，可以使用位运算和逻辑操作来控制字符的位置和显示。

1. 通过使用单片机控制595器件的MR管脚和OE管脚可以实现以下功能：

（1）功能扩展：通过单片机控制可以实现更多的功能。例如，根据特定条件或输入信号来动态地控制MR管脚和OE管脚，从而改变595器件的行为，可以根据需要实现各种功能，如数据的存储、显示、时序控制等。

（2）时间控制：可以提供精确的时间控制，可以控制MR管脚和OE管脚的状态持续时间和时序。这对于确保正确的数据传输和显示非常重要。