**一、实验目的与要求**

1、熟悉8255的功能，了解点阵显示的原理及控制方法；

2、学会使用LED点阵，通过编程显示不同字符；

3、认真预习本节实验内容，可尝试自行编写程序，做好实验准备工作，填写实验报告。

**二、实验设备**

STAR系列实验仪一套、PC机一台。

**三、实验内容**

1、编写程序，用8255的PA、PB控制16X16点阵的行；8255的PC口、8155的PB口控制16X16点阵的列；显示字符。

2、按图连接线路；运行程序，观察实验结果，学会编程控制LED点阵显示字符。

**四、实验原理图**

## A2区：16×16 LED实验电路



JP14、JP15组成16根行扫描线；JP21、JP22组成16根列扫描线。

## A3区：CPU总线、片选区

|  |  |
| --- | --- |
| JP28：地址线A0..A7；  JP32：地址线A8..A15； |  |
| JP29：地址线A0..A9；  JP33：地址线A8..A17； | 根据数据总线宽度，选择合适的地址线 |
| JP41、JP42：数据总线D0..D7 | |
| JP39、JP40：数据总线D8..D15 | |
| JP47、JP48：数据总线D16..D23 | |
| JP45、JP46：数据总线D24..D31 | |
| 控制线：IOR、IOW、MEMR、MEMW、HOLD、HLDA、BLE、BHE、INTR、INTA | |
| BK1、BK2：备用 | |

**片选区**

|  |  |  |
| --- | --- | --- |
| 片选 | 地址范围 | 说明 |
| mCS0 | 80000H～BFFFFH | 存贮器芯片的片选，16位数据总线 |
| CS1 | 0270H～027FH | I/O芯片的片选，8位数据总线 |
| CS2 | 0260H～026FH |
| CS3 | 0250H～025FH |
| CS4 | 0240H～024FH |
| CS5 | 0230H～023FH | I/O芯片的片选，16位数据总线 |

**五、实验步骤**

1、主机连线说明：

|  |  |  |
| --- | --- | --- |
| D3区：CS（8255）、A0、A1 | —— | A3区：CS1、A0、A1 |
| D3区：JP23(PA)、JP20(PB) | —— | A2区：JP21、JP22（列输出线） |
| B4区：JP57(D0..D7) | —— | A3区：JP42(D0..D7) |
| B4区：JP56(D8..D15) | —— | A3区：JP40(D8..D15) |
| B4(I/O)区：CS273、BLE、BHE | —— | A3区：CS5、BLE、BHE |
| B4(I/O)区：RD、WR | —— | A3区：IOR、IOW |
| B4(I/O)区：JP51、JP55 | —— | A2区：JP14、JP15（行输出线） |

**（注意连线方向）**

2、运行程序，观察实验结果。运行演示程序将会看到字符“欢迎使用星研实验仪”在点阵上自下而上循环移动显示。

**六、****实验扩展及思考**

1、修改程序，使显示的字符从左至右动态循环显示。

先定义各个IO口和控制口地址

.MODEL TINY

ADDR\_8255\_PA EQU 0270H ;8255 PA口;列线控制端口1

ADDR\_8255\_PB EQU 0271H ;8255 PB口;列线控制端口2

ADDR\_8255\_C EQU 273H ;8255控制口

ADDR\_273 EQU 230H ;8155 PA口;行线控制端口

LINE EQU ADDR\_273 ;行控制线

ROW1 EQU ADDR\_8255\_PA ;列控制线1

ROW2 EQU ADDR\_8255\_PB ;列控制线2

输入16\*16点阵字形码

由二极管组成，当AB端电压差（ROW=1,LINE=0）大于一定值时二极管导通，LED发亮（加上LED原理图）

字形码

每两个字节构成一行，输入列控制线ROW1和ROW2，共16行

NONE为全零，由于字形码自下而上滚动，需要把最后一个字流动完，所以需要一个32字节的全零数据。

首先用INIT\_IO函数往8255输入控制字，设置8255的PA、PB口为输出口

再用测试函数TEST\_LED测试LED是否可以全亮

最后用CLEAR清屏，使LED全灭

将需要滚动的字符计数——CX1

将SI原地址指针指向字形码数据首地址

将一个完整字符需要滚动的次数（16屏）计数——CX2

利用LED的刷新控制点阵的滚动速度

进入显示一个完整字符的子程序

一屏刷新次数（控制滚动频率）——CX3

将显示一屏的全部行数（16）给计数器——CX4

将IO扩展口输入的最低位输入0，其他位置1（LINE=FFFEH）

①把一行的字形码送入ROW1和ROW2：用LODSB读入当前字节码的一个字节到PA数据口（ROW1），再读入下一个字节到PB数据口（ROW2），若LED的列控制线引脚与PA、PB口的位号相反，则需要调用ADJUST函数将二进制数旋转180°（例：D7~D0 -> D0~D7）

②将LINE数据循环左移（RCL）例：LINE=FFFEH->FFFDH;

依次执行①、②共16次，每次CX3计数减一，显示一个完整的字形码

当显示完一屏点阵后SI+2，使得下一次显示的点阵为当前点阵向上滚动一行，并将CX1减一，若CX1=0则已将所有字形码滚动完毕，否则继续循环

将所有字符滚动完毕后用JMP指令跳转到显示第一个字符的代码，循环执行字符滚动

扩展将字形码从左到右输出：

改变了字节码为“我爱微机”

相比于上一个实验主要改变了显示一个完整字符的子程序代码

具体改动为：

将ROW2的最低位输入1，ROW2的其他位以及ROW1所有位清0（ROW2=01H,ROW1=00H）

①把一行的字形码送入LINE：用LODSB连续读入当前字节码的两个个字节到LINE，若LED的行控制线引脚与I/O扩展口的位号相反，则需要调用ADJUST函数将二进制数旋转180°（例：D7~D0 -> D0~D7）

②将LINE数据循环左移（RCL）例：LINE=FFFEH->FFFDH;

依次执行①、②共16次，每次CX3计数减一，显示一个完整的字形码