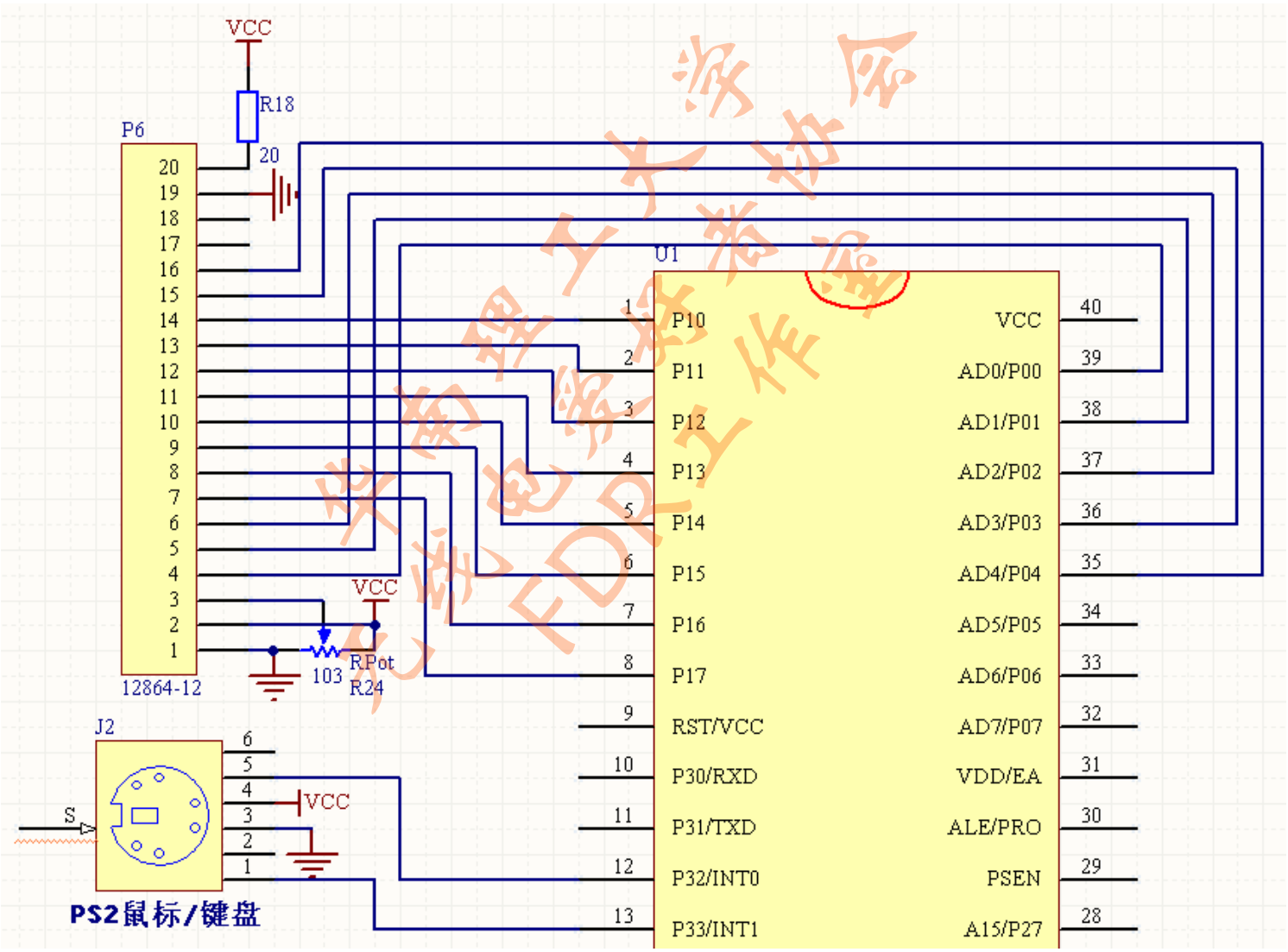
实验二十一:PS2键盘控制LCD12864实验

一. 实验目的:

1. 认识ps2键盘协议
2. 掌握12864液晶模块的使用方法。
3. 学会利用12864实现小型可视化图形界面

二. 实验原理:



12864 与单片机连接原理图

12864 跟 ps2 键盘的底层通讯较为复杂，我们在文件夹里提供了相应的资料给读者。在此不再累述。提醒读者一下：12864 驱动电流加大，使用时请先调 lcd 座旁的蓝色可调电阻，使 lcd 上能显示一个一个的小方格。然后才能正常显示。12864 有并行跟串行两种通讯方式。当 psb 脚为高平时是并行，反之为串行。

键盘要注意通码（按下键盘时发出）跟断码（松开键盘时发出）之间的联系。

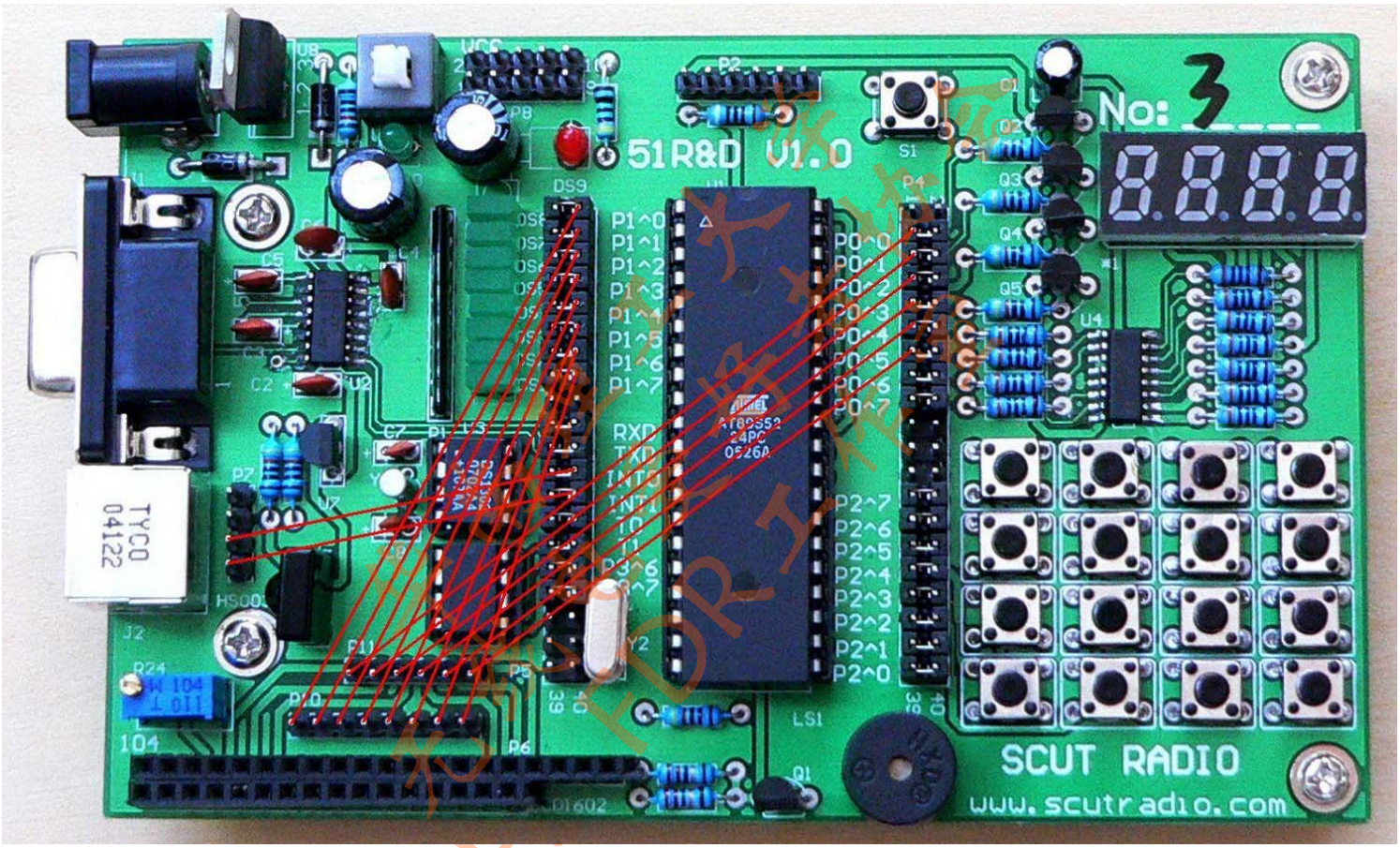
三．硬件链接图：

12864连线：

如图所示的红线排针连到单片机各IO口，12864的几个控制脚都是连接在P0口上，从右到左依次是RS、RW、E、PSB、NC(我们提供的12864用不到该引脚)、RST，依次连接到P0^0,P0^1,P0^2,P0^3,P0^4,P0^5;而数据线则直接接到P1口。

键盘连线：

键盘的两条控制线在排针P7靠近红外接收头的两根上，最后一根是时钟线sclk，上边一根是数据线dat，将数据线接到单片机P3^2上靠近单片机一侧的排针上，将时钟线接到单片机INT1上。



12864连线图

四.实验程序：

```
#include<AT89X52.h>
#include"led.h"  //该头文件包含数码管显示数字驱动
#include"tg12864p.h"//该头文件包含 lcd12864 驱动

sbit dat =P3^2;//DATA IN
```

```
#define uchar unsigned char //  即  uchar 等同与 unsigned char
#define uint unsigned int
uchar sp2key_scan();
/*****************常量声明区*****************************************/
uchar code key_table[47]={
                            '0','1','2','3','4','5','6','7','8','9',
                            'a','b','c','d','e','f','g','h','i','j',
                          'k','l','m','n','o','p','q','r','s','t',          //说明：存放键盘上按键的 as2 码值
                          'u','v','w','x','y','z',' ','-','=',']',
                              ';',"",',','.','/','[',' '                    //作用：方便日后传值到液晶上
                           };

 /****************全局变量声明区*************************************/
uchar key=0;          //用于存放键盘解码函数的解码结果
uchar key_temp=0;
bit     BF=0;         //接收到通码的标志位
uchar key_count=0; //记录键盘传任一个数据时传来时钟脉冲个数
uchar key_buffer[5]={0,0,0,0,0};//接收键盘传来数据的缓冲区
uchar key_num=0;//一个按键动作，键盘会发出通码，段吗，这用于记录收到编码个数。
                //可以参考附带的 ps2.pdf 第 54 页

/*************          键盘接收解码程序      ****************************
这是本程序的核心函数，用于键盘解码。建议读者参考附带的 pse.pdf 第 54 页
了解键盘 ps2 协议，理解通码、断码概念以及发送编码个数。
*******************************************************************/
void    key_dec() interrupt 2 using 0
{
    if ((key_count>0)&&(key_count<9))
    {
        key_temp>>=1;
        if (dat==1)key_temp|=0x80;
    }
    key_count++;
    if (key_count>10)
    {
        key_buffer[key_num]=key_temp; //将获得的数据写入缓冲区
        key_temp=0;
        key_num++;
        key_num%=5;             //不会超过五个码
        if(key_buffer[0]<0xe0) //收到通码或结束码
        {
            if(key_num==1) //若是通码
            {
```

```
                if(!BF)
                {      //置标志位
                    BF=1; //清空缓冲区，准备接收下一个按键发来的码
                    key=key_buffer[0];
                }
            }
            if(key_num>1&&key_buffer[key_num-1]==key_buffer[0])key_num=0;   //收到结束码
        }
        else
        {
            if(key_num==2) //收到通码或结束码
            {
                if(!BF) //若是通码
                {                   //置标志位若置标志位是通码
                    BF=1;       //清空缓冲区，准备接收下一个按键发来的码
                    key=key_buffer[1];
                }
            }
            if(key_num>2&&key_buffer[key_num-1]==key_buffer[1])key_num=0;//收到结束码
        }
        key_count=0;
    }
}

/****************     键盘初始化函数  ***********************/

void init_keyboard()
{
    TCON=0x04; //外部中断低电平触发
    IE   =132;     //开启外部中断 1
                //不修改当前的优先级
}

  /**********************          键值转换函数     **********************
  主要用于将接收到的编码量化键值，以备以后分类处理。
  返回数据：     量化后的键值
  ***************************************************************************/


uchar sp2key_scan()
{
     unsigned char temp2,k=255;
     if(BF==1)          //如果收到
     {                  //清除接收到通码的标志位，键盘可以继续接收数据
```

华南理工学生会 无线电爱好者协会 FDR工作室

```
 BF=0;
       temp2=key;
 switch ( temp2 )
{
      case 0x45: k=0;    break; //0
      case 0x16: k=1;    break; //1
      case 0x1e: k=2;    break; //2
      case 0x26: k=3;    break; //3
      case 0x25: k=4;    break; //4
      case 0x2e: k=5;    break; //5
      case 0x36: k=6;    break; //6
      case 0x3d: k=7;    break; //7
      case 0x3e: k=8;    break; //8
      case 0x46: k=9;    break; //9
      case 0x1c: k=10; break; //a
      case 0x32: k=11; break; //b
      case 0x21: k=12; break; //c
      case 0x23: k=13; break; //d
      case 0x24: k=14; break; //e
      case 0x2b: k=15; break; //f
      case 0x34: k=16; break; //g
      case 0x33: k=17; break; //h
      case 0x43: k=18; break; //i
      case 0x3b: k=19; break; //j
      case 0x42: k=20; break; //k
      case 0x4b: k=21; break; //l
      case 0x3a: k=22; break; //m
      case 0x31: k=23; break; //n
      case 0x44: k=24; break; //o
      case 0x4d: k=25; break; //p
      case 0x15: k=26; break; //q
      case 0x2d: k=27; break; //r
      case 0x1b: k=28; break; //s
      case 0x2c: k=29; break; //t
      case 0x3c: k=30; break; //u
      case 0x2a: k=31; break; //v
      case 0x1d: k=32; break; //w
      case 0x22: k=33; break; //x
      case 0x35: k=34; break; //y
      case 0x1a: k=35; break; //z
      case 0x0e: k=36; break; //~
      case 0x4e: k=37; break; //-
      case 0x55: k=38; break; //=
      case 0x29: k=46; break; //SPACE
```

华南理工学生电子爱好者协会 FDR工作室

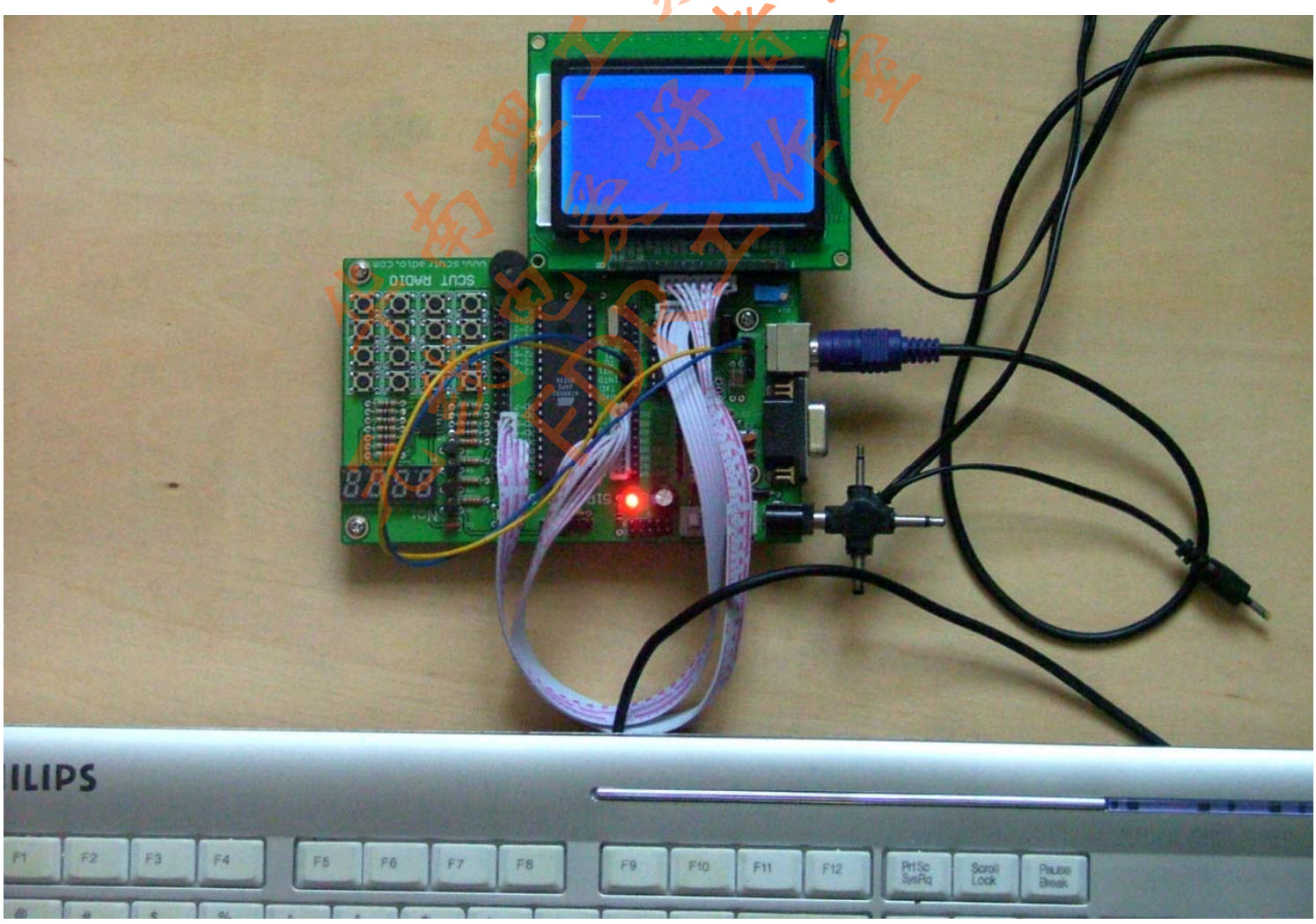
```
                case 0x66: k=47;break; //del
                case 0x5b: k=39;break; //]
                case 0x4c: k=40;break;//;
                case 0x52: k=41;break;// '
                case 0x41: k=42; break; //,
                case 0x49: k=43;break;//.
                case 0x4a: k=44;break; // /
                  case 0x54: k=45; break; //[
               case 0x58: k=49; break; //cap
               case 0x5a: k=48; break; //回车键

               case 112 : k=0 ; break; //小键盘部分
               case 105 : k=1 ; break;
               case 114 : k=2 ; break;
               case 122 : k=3 ; break;
               case 107 : k=4 ; break;
               case 115 : k=5 ; break;
               case 116 : k=6 ; break;
               case 108 : k=7 ; break;
               case 117 : k=8 ; break;
               case 125 : k=9 ; break;
               case 113 : k=43 ; break;;

                default : k=255;
          }

     }
     return k;
}
/************************** 输入显示函数*************************************
                        将输入数据显示在 12864 上
                    data  输入字符的 asc2 码值，
                    positon     是当前坐标
*****************************************************************************/
void input(uchar dat,uchar position)
{
     switch(position)
     {
          case 16:Location_xy_12864(1,0); break;
          case 32:Location_xy_12864(2,0); break;
          case 48:Location_xy_12864(3,0); break;
     }
     Write_Char_12864(dat);
}
```

```
void   main()
{
    uchar t=0,
    temp=0,
    pos=0,  //记录输入光标位置，输入一个字母或数字，光标加一
    cap=0;
    temp=200;
    while(temp--);
    init_keyboard();
    Init_12864();
    delay10us(1000);
    while(1)
    {
        temp=sp2key_scan();
        if(temp!=255)
        {
            t=temp;
            if(t<48) //输入为字母或其他符号
            {
                if(pos==64)
                {
                    pos=0;
                    LcmClearTXT();
                }
                if(t==47)                   //删除
                {
                    if(pos)
                    {
                        Lcm_clear_char(pos);
                        if((pos%2))pos-=1;
                        if(pos)pos--;
                    }
                }
                else
                {
                    temp=key_table[t];
                    if(t>9&&t<36)temp-=cap*32; //如果输入为字符,根据 cap  令显示为大写或小写
                    input(temp,pos);
                    pos++;
                }
            }
            if(t==48) //换行
            {
                if(pos<48)
```

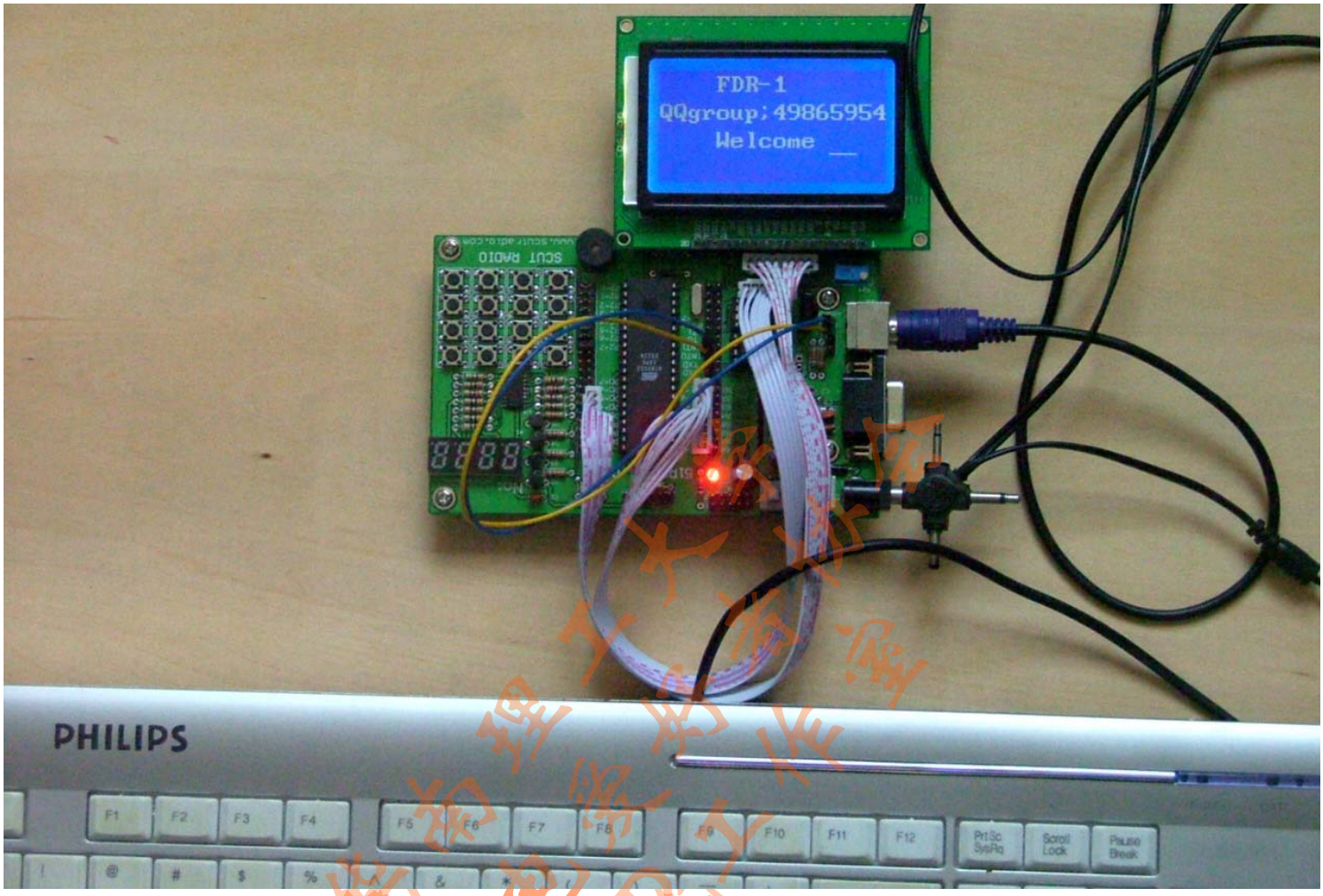
华南理工学生电子爱好者协会 无线 FDR工作室

```
                {
                    pos+=16;
                    Location_xy_12864(pos/16,(pos%16)/2);
                }
                else
                {
                    LcmClearTXT();
                    pos=0;
                }
            }
            if(t==49)cap=!cap;//cap：大小写转换
        }
    }
}
```

五．实际效果图 ：



图片一



图片二

华南理工大学电子与信息学院 无线电协会 FDR工作室