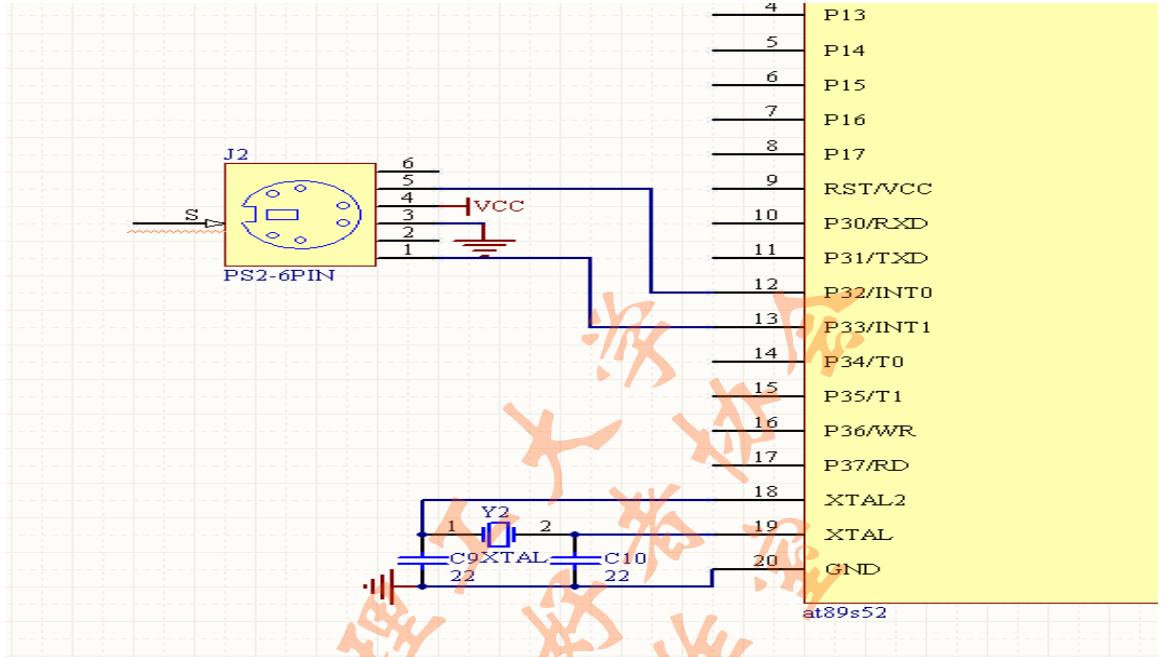


## 实验二十：LCD1602+PS2 键盘输入实验

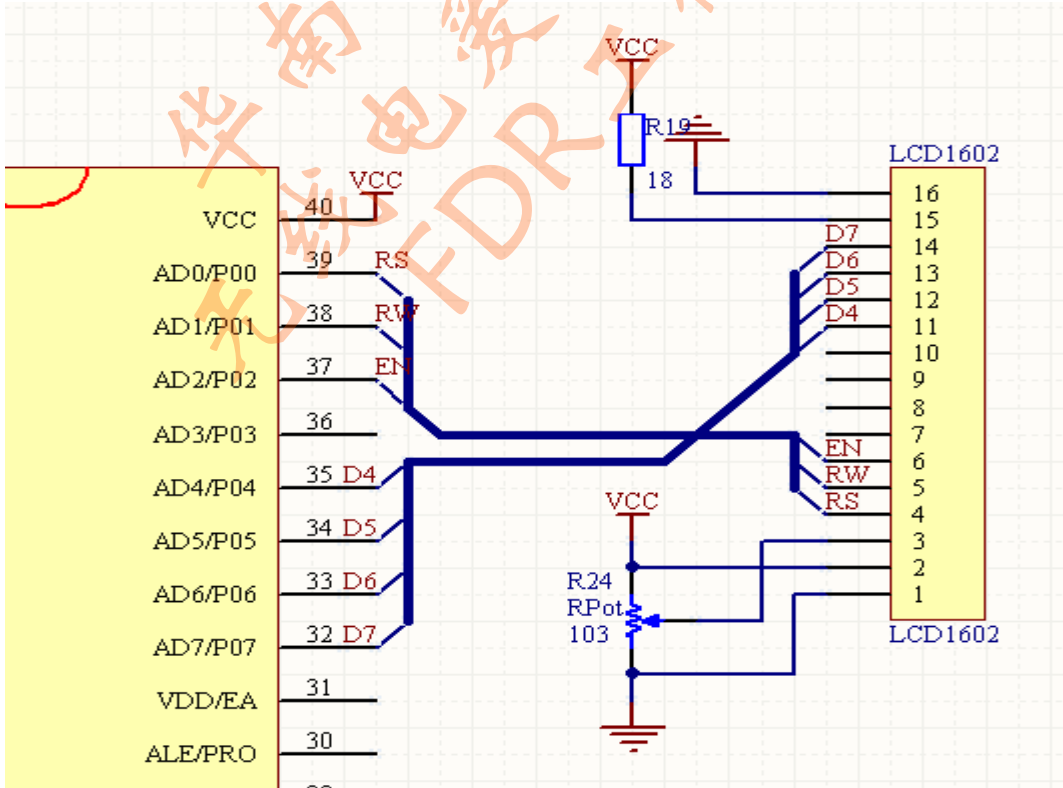
### 一.实验目的:

熟悉 PS2 协议。

### 二.实验原理图:

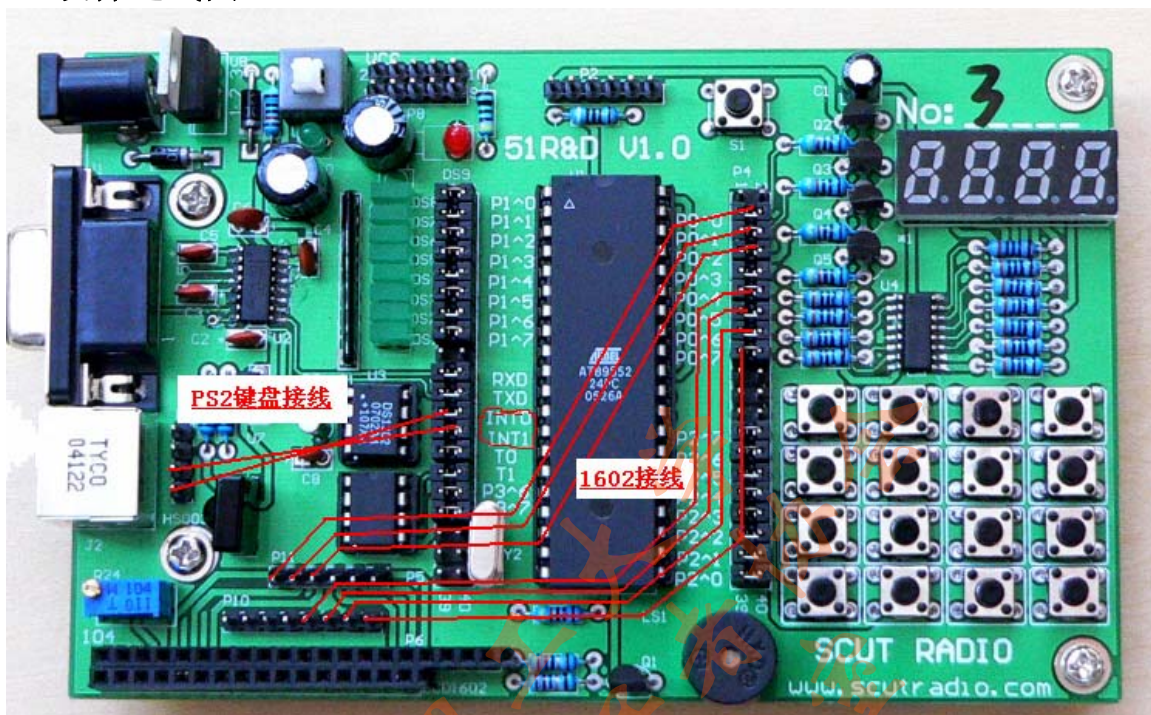


键盘与单片机的连接



LCD1602 连接图

### 三.硬件连线图:



### 四.实验原理:

PS/2 通信协议是键盘与其他硬件通信时的一种半双工通信协议，键盘通过 Data 线向外发送数据，数据时钟由键盘产生，在协议中数据的传送格式为短帧格式，一般为 11 位，各位的含义如下图所示：

Start	D0	D1	D2	D3	D4	D5	D6	D7	Parity	Stop
-------	----	----	----	----	----	----	----	----	--------	------

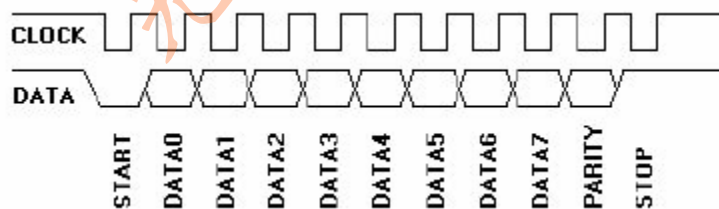
Start: 起始位，总为 '0' (低电平)

D0~D7: 8 位数据位 (低位在前，高位在后)

Parity: 奇偶校验位 (为奇校验)

Stop: 停止位，总为 '1' (高电平)

键盘向外发送数据的时序图如下图所示：



键盘首先检查 Clock 线和 Data 线是否同时为高电平时，若是，则键盘产生 Clock 信号，并将待发的数据按照数据帧的格式逐位放在数据线上，数据线在时钟的高电平期间改变状态，时钟的下降沿将数据锁定到数据线上，此时外界就可以到数据线上读取数据。键盘同步时钟的最大频率为 33KHZ，一般情况下使用 10~20KHZ 的频率。

本实验主要是接收由键盘 Data 线发送过来的 11 位串行数据。用的是 11 位移位寄存器的基本原理。实现将由键盘发送过来的键盘扫描码转换成 ASCII 码，并在 LCD1602 液晶上显示出来。

### 五.实验代码:

```
//=====ps2.h 头文件=====//
#ifndef PS2_H
#define PS2_H

sbit keydata=P3^3;
sbit clk=P3^2;

unsigned char times=0;
unsigned char i=0;
unsigned char keycode=0,ps2_key; //ps2_key 用于存放接收到的键码

static unsigned char BF=0; //标识是否有字符被收到

unsigned char code noshift[80][2]=
{
  1, 8, // { f9 }
  3, 4, // { f5 }
  4, 2, // { f3 }
  5, 0, // { f1 }
  6, 1, // { f2 }
  7, 11, // { f12 }
  9, 9, // { f10 }
  13, 25, // { tab }
  20, 27, // { ctrl }
  17, 28, // { alt }
  41, 29, // { space }
  31, 30, // { win }
  12, 3, // { f4 }
  11, 5, // { f6 }
  10, 7, // { f8 }
  14, 96, // { ` }
  22, 49, // { 1 }
  28, 97, // { a }
  30, 50, // { 2 }
  33, 99, // { c }
  38, 51, // { 3 }
  37, 52, // { 4 }
  46, 53, // { 5 }
  47, 31, // { winright }
  54, 54, // { 6 }
  61, 55, // { 7 }
  62, 56, // { 8 }
  50, 98, // { b }
  35, 100, // { d }
  36, 101, // { e }
  43, 102, // { f }
  52, 103, // { g }
  51, 104, // { h }
  59, 106, // { j }
  58, 109, // { m }
  49, 110, // { n }
  21, 113, // { q }
  45, 114, // { r }

```

```
27,115, // { s }
44,116, // { t }
60,117, // { u }
42,118, // { v }
29,119, // { w }
34,120, // { x }
53,121, // { y }
26,122, // { z }
65,44, // { , }
66,107, // { k }
67,105, // { i }
68,111, // { o }
69,48, // { 0 }
70,57, // { 9 }
73,46, // { . }
74,47, // { / }
75,108, // { l }
76,59, // { ; }
77,112, // { p }
78,45, // { - }
82,39, // { ' }
85,61, // { = }
84,91, // { [ }
91,93, // { ] }
88,26, // { caps }
93,92, // { \ }
90,32, // { enter }
120,10, // { f11 }
102,12, // { back }
224,13, // { home }
105,14, // { end }
125,15, // { pageup }
122,16, // { pagedown }
117,17, // { up }
114,18, // { down }
107,19, // { left }
116,20, // { right }
113,21, // { del }
112,22, // { insert }
225,23, // { pause }
118,24, // { esc }
131, 6, // { f7 }
};
```

```
unsigned char code addshift[47][2]=
```

```
{
14,126, // { ~ }
22, 33, // { ! }
30, 64, // { @ }
38, 35, // { # }
37, 36, // { $ }
46, 37, // { % }
54, 94, // { ^ }
```

```
61,38, // { & }
62,42, // { * }
70,40, // { ( }
69,41, // { ) }
78,95, // { _ }
85,43, // { + }
93,124, // { | }
84,123, // { { }
91,125, // { } }
76,58, // { : }
82,34, // { " }
65,60, // { < }
73,62, // { > }
74,63, // { ? }
28,65, // { a }
50,66, // { b }
33,67, // { c }
35,68, // { d }
36,69, // { e }
43,70, // { f }
52,71, // { g }
51,72, // { h }
67,73, // { i }
59,74, // { j }
66,75, // { k }
75,76, // { l }
58,77, // { m }
49,78, // { n }
68,79, // { o }
77,80, // { p }
21,81, // { q }
45,82, // { r }
27,83, // { s }
44,84, // { t }
60,85, // { u }
42,86, // { v }
29,87, // { w }
34,88, // { x }
53,89, // { y }
26,90, // { z }
};
```

```
unsigned char getchar(unsigned char k) //转换键码为 ASCII 码
{
    unsigned char j;
    if(!i)
        for(j=0;j<80;j++)
        {
            if(noshift[j][0]==k)
                {ps2_key=noshift[j][1];return 1;}
        }
    else
        for(j=0;j<47;j++)
```



```

    {
        if(addshift[j][0]==k)
            {ps2_key=addshift[j][1];return 1;}
    }
    return 0;
}
void Keyboard_out(void) interrupt 0
{
    if(times<9)
    {
        keycode=keycode>>1;           //因键盘数据是低>>高，结合上一句所以右
移一位
        if(keydata) keycode=keycode | 0x80; //当键盘数据线为 1 时为 1 到最高位
    }
    times++;
    if(times>10)
    {
        times=0;
        if(keycode==0xe0 || keycode==0xf0){}return;}
        else if((keycode==18 || keycode==89) && i==0)
            {i=1;}
            else if((keycode==18 || keycode==89) && i==1)
            {i=0;}
            else {EX0=0;BF=1;} //关中断等显示完后再开中断
//当中断 11 次后表示一帧数据收完，清变量准备下一
次接收
//（注：如这里不用 BF 和关中断直接调 Decode()
//则所 Decode 中所调用的所有函数要声明为再入函
数)
    }
    // while(!clk); //等待 PS/2CLK 拉高
}
#endif
//=====1602 液晶头文键=====
#ifndef LCD1602_4_H
#define LCD1602_4_H

#include <reg52.h>
#include <intrins.h>

#define MSB      0x80      //读忙位
#define HIGH     1
#define LOW      0

/*****
#define DATA_MODE      0x28    //LCD 模式设置，0x28 位四位数据线模式
#define OPEN_SCREEN     0x0f    //打开 LCD 显示,有光标；0x0c 没有光标
#define DISPLAY_ADDRESS 0x80    //写地址指令
#define CLEARSCREEN     LCD_en_com(0x01) //清屏

```

```

/*****//LCD1602 的接线*****/
#define LCDIO    P0    //LCD1602 的高四位接在 P0 口的高四位

sbit LCD1602_RS=P0^0;    //RS 为寄存器选择,高电平时选择数据寄存器、低电平时选择指令寄
存器。
sbit LCD1602_RW=P0^1;    //RW 为读写信号线, 高电平时进行读操作, 低电平时进行写操作。
sbit LCD1602_EN=P0^2;    //E 端为使能端, 当 E 端由高电平跳变成低电平时, 液晶模块执行命
令。

/*****
void LCD_en_com(unsigned char command);                //写命令指
令
void LCD_en_dat(unsigned char temp);                  //写数据指令
void LCD_set_xy(unsigned char x, unsigned char y);    //设置显示地址:X 为
横轴(0-15), Y 为纵轴(0-1).
void LCD_write_char(unsigned x,unsigned char y,unsigned char dat); //写一个字符
void LCD_write_string(unsigned char x,unsigned char y,unsigned char *s);//写一个字符串
void LCD_write_int(unsigned char x,unsigned char y,unsigned int intdat); //写无字符型整形数
void LCD_init(void);                                  //lcd 初始化
void LCD_Read_BF(void);                               //LCD 读忙
信号
*****/

/*
LCD1602.c

Created by Zhengxihui
email: zhengxihui0506@126.com
May 2007

*/

void LCD_Read_BF(void)
{
    LCD1602_RW=HIGH;    //RW 1
    LCD1602_RS=LOW;     //RS 0
    LCD1602_EN=HIGH;    //EN 1 读忙信号
    LCDIO=LCDIO&0x0F|0xf0;
    while(LCDIO&MSB);
    LCD1602_EN=LOW;
}

void LCD_en_write(void)    //EN 端产生一个高电平脉冲, 写 LCD
{
    LCD1602_EN=HIGH;
    _nop_();
    LCD1602_EN=LOW;
}

/*****
void LCD_en_com(unsigned char command)

```



```
{
    LCD_Read_BF();
    LCD1602_RS=LOW;      //RS 0 写命令
    LCD1602_RW=LOW;      //RW 0

    LCDIO&=0x0F;
    LCDIO=command&0xf0 | LCDIO&0x0f; //先发送高四位数据
    LCD_en_write();
    command=command<<4;
    LCDIO&=0x0F;
    LCDIO=command&0xf0 | LCDIO&0x0f; //发送低四位数据
    LCD_en_write();
}
//*****
void LCD_en_dat(unsigned char dat)
{
    LCD_Read_BF();
    LCD1602_RS=HIGH;     //RS 1 写数据
    LCD1602_RW=LOW;      //RW 0
    LCDIO &=0x0F;
    LCDIO=dat&0xf0 | LCDIO&0x0f; //先发送高四位数据
    LCD_en_write();
    dat=dat<<4;
    LCDIO &=0x0F;
    LCDIO=dat&0xf0 | LCDIO&0x0f; //发送低四位数据
    LCD_en_write();
}
//*****设置显示地址:X为横轴(0-15), Y为纵轴(0-1)*****
void LCD_set_xy(unsigned char x,unsigned char y)
{
    unsigned char address;
    x&=0x0f;
    if(!y)
        address=0x80+x;
    else
        address=0xc0+x;
    LCD_en_com(address);
}
//*****写一个字符: X为横轴(0-15), Y为纵轴(0-1)*****
void LCD_write_char(unsigned x,unsigned char y,unsigned char dat)
{
    LCD_set_xy(x,y);
    LCD_en_dat(dat);
}
//*****写一个字符串*****
void LCD_write_string(unsigned char x,unsigned char y,const unsigned char *s)
{
    LCD_set_xy(x,y);
    while(*s)
    {
        LCD_en_dat(*s);
        s++;
    }
}
```



```
    }
}
//*****写一个无字符整数*****
void LCD_write_int(unsigned char x,unsigned char y,unsigned int intdat)
{
    unsigned char i=0,temp[5];
    while(intdat/10 || intdat%10)
    {
        temp[i]=intdat%10;
        intdat/=10;
        i++;
    }
    LCD_set_xy(x,y);
    while(i)
    {
        i--;
        LCD_en_dat(temp[i]+0x30);
    }
}

//*****//LCD 的初始化函数(刚上电的时候一定要调用一次)*****
void LCD_init(void)
{
    LCD_en_com(DATA_MODE); //设计四位数据线模式
    LCD_en_com(DATA_MODE); //设计四位数据线模式
    LCD_en_com(OPEN_SCREEN); //打开 LCD 显示
    LCD_en_com(DISPLAY_ADDRESS); //设计首地址
    CLEARSCREEN; //清屏
}

#endif
//=====主函数文件=====
#include <reg52.h>

#include "LCD1602_4.h"
#include "ps2.h"

unsigned char key_bit=0,key_mem[10]={0};

void main()
{
    unsigned char temp=0,i=0,j=0,y=0;
    LCD_init(); //1602 初始化
    EA=1; //开总中断
    IT0=1;
    EX0=1; //开 INT0 中断
    LCD_write_string(0,0,"Ready Go!"); //说明初始化成功

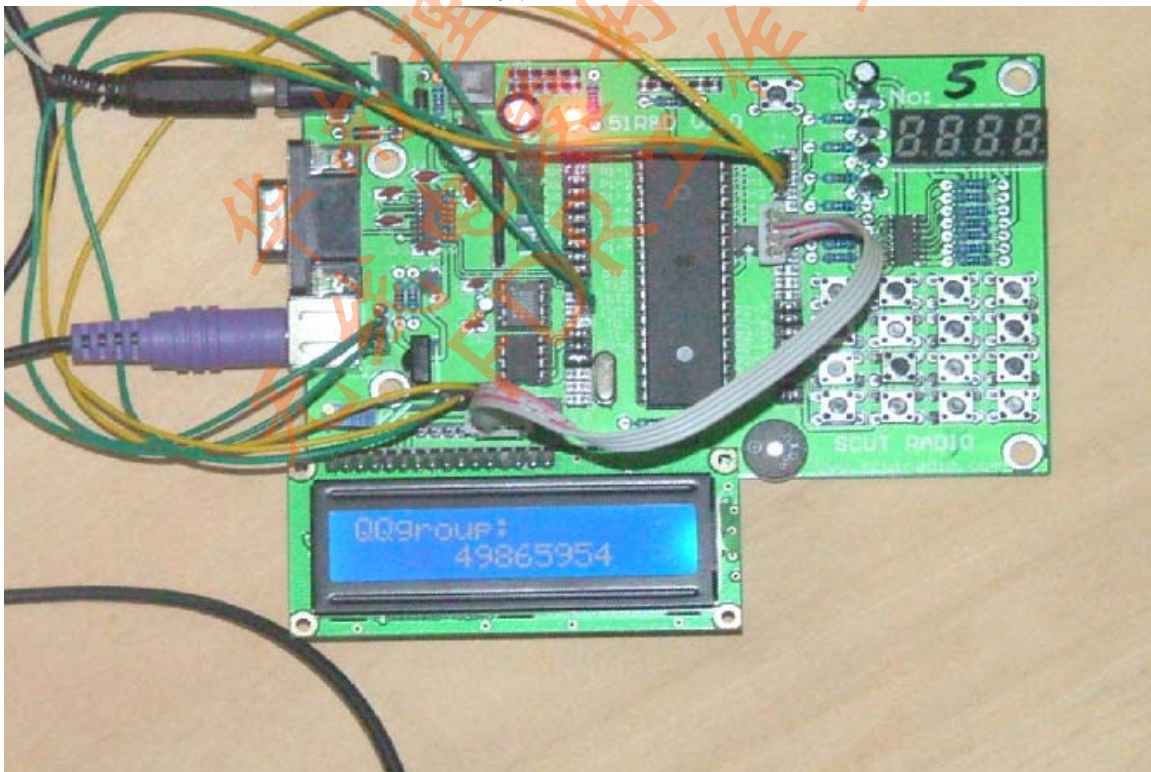
    while(1)
    {
        if(BF) //如果有按键按下则放进缓冲区
        {
```

```
    j+=getchar(keycode);    //去掉断码
    if(j==2)
    {
        key_mem[key_bit]=ps2_key;//放进缓冲器
        key_bit++;
        j=0;
    }
    BF=0;                    //再开中断
    EX0=1;
}
while(key_bit!=0)          //缓冲区如果有键码，则显示
{
    if(temp>0x0f && keycode!=102)//液晶换行
    {
        temp=0;
        if(y==0)y++;
        else {y=0;CLEARSCREEN;} //显示满屏幕则清屏
    }
    if(keycode==102)        //如果有 BackSpace 按下则删掉前一个键
    {
        temp--;
        LCD_write_char(temp,y,' ');
        LCD_set_xy(temp,y);
    }
    else
    {
        LCD_write_char(temp,y,key_mem[0]);//显示缓冲区的键码
        temp++;
    }
    for(i=0;i<key_bit;)
    {
        key_mem[i]=key_mem[i+1];
        i++;
    }
    key_bit--;
    if(BF==1)break;
}
}
```

六.效果图:



图一



图二



图三

华南理工大学  
无线电爱好者俱乐部  
FDR工作室