

按照模式分：自动模式，手动模式

开机时蜂鸣器响一声，通过蓝牙向手机发送当前空气湿度，并进入**自动模式**

自动模式：LCD 实时显示当前湿度、当前模式(A)以及当前晾衣架状态(IN/OUT)，在湿度大于 80（资料中说法不一，有的 70 有的 90，这里我们将 80 定为晾收衣服的阈值）时或在光照小于设定值时（夜晚），电机反转且 LCD 显示 ‘IN’，碰到限位开关时停止反转，蜂鸣器响三声，限位指示灯闪烁三次，晾衣架收回；在湿度小于 80 时且光照大于设定值时（白天），电机正转且 LCD 显示 ‘OUT’，碰到限位开关时停止正转，蜂鸣器响三声，限位指示灯闪烁三次，晾衣架伸出

长按模式转换按钮可在两种模式之间转换

手动模式：LCD 实时显示当前湿度以及当前模式（M），手动控制晾衣架的收回和伸出。按下反转键后，蜂鸣器响一声，电机反转且 LCD 显示 ‘IN’，碰到限位开关时停止反转，蜂鸣器响三声，限位指示灯闪烁三次，晾衣架收回；按下正转键后，蜂鸣器响一声，电机反转且 LCD 显示 ‘OUT’，碰到限位开关时停止正转，蜂鸣器响三声，限位指示灯闪烁三次，晾衣架伸出。晾衣架收回光照设定值可由用户通过旋钮设置

具体功能及实现

由于这次的项目功能较多且会相互影响，于是我们选择了更利于模块化的编程语言——C 语言

蜂鸣器：利用 PWM 改变音量

```
void Simulate_Pwm(unsigned char period, unsigned char duty)
{
    int i;
    unsigned char low = period / 100 * duty; //低电平延时时间
    unsigned char high = period - low;       //高电平延时时间
    for(i=0;i<=500;i++)
    {
        beep = 0;           //蜂鸣器IO电平置低
        Delay_10us_1(low);  //低电平延时时间
        beep = 1;           //蜂鸣器IO电平置高
        Delay_10us_1(high); //高电平延时时间
    }
}

void buzz(ui number)
{
    if (number==1)
    {
        Simulate_Pwm(100, 99); //周期100*10us, 占空比50%—频率: 1KHz
        beep=0;
    }
    else
    {
        beep=0;
    }
}
```

蓝牙

发送数据

```
void SendData(int humidity)
{
    int i=0;
    unsigned char gewei,shiwei,array1[]={"HUMIDITY:  "};
    gewei=humidity%10+'0';
    shiwei=humidity/10+'0';
    array1[9]=shiwei;
    array1[10]=gewei;

    while(array1[i] != '\0')
    {
        SBUF=array1[i];
        while(!TI); //若没有发送完数据 等待
        TI=0; //若发送完 标志位清0
        i++;
    }
}
```

接收数据

```
void SeriesInterrupt() interrupt 4
{
    if(RI) //接收完毕 接收中断位RI置1
    {
        DataGet=SBUF; //取出接收的数据
        RI=0;
        DataGet=DataGet-'0';
        SendData2(DataGet); //将接收的数据发送出去
        // if(DataGet=='Y')
        //     P1=0x00; //开灯
        // else if(DataGet=='N')
        //     P1=0xFF; //关灯
        // else
        //     P1=0x55; //交替
    }
}
```

LCD 显示

写命令:

```
void write_com(unsigned char command)
{
    RS = 0;
    RW = 0; //高读低写
    E=0;
    LCD = command;
    delay5ms(); //这里延时最低要30纳秒 我们直接给5ms
    E = 1; //使能拉高
    delay5ms(); //最低要求延迟150纳秒 我们直接给5ms
    E = 0;
}
```

写数据:

```
void write_data(unsigned char dat)
{
    RS = 1;
    RW = 0;
    E=0;
    LCD = dat;
    delay5ms(); //这里延时最低要30纳秒 我们直接给5ms
    E = 1; //使能拉高
    delay5ms(); //最低要求延迟150纳秒 我们直接给5ms
    E = 0;
}
```

初始化:

```

void init_lcd1()
{
    write_com(0x06); //写入数据后光标自动右移 整屏不移动。
    write_com(0x0c); //开显示功能 无光标 不闪烁
    write_com(0x38); //数据总线8位 16X2显示 5*7点阵
    write_com(0x01); //清屏 0000 0001
    write_com(0x81); //在1602第一行第1个开始显示
}

```

显示静态字符：

```

void display(void)
{
    uc array1[]={"HUMIDITY:      00000000000000000000000000000000 MODE:"};
    while(array1[i] != '\0')
    {
        write_data(array1[i]);
        i++;
    }
}

```

显示动态字符：如湿度

```

write_com(0x8A);
testnum = RH();
FlagStartRH = 0;
humidity = U8RH_data_H; //读出湿度，只取整数部分
gewei=humidity%10+'0';
shiwei=humidity/10+'0';
write_data(shiwei);
write_com(0x8B);
write_data(gewei);

```

温湿度传感

```

void COM(void)
{
    uc i;
    for(i=0;i<8;i++)
    {
        U8FLAG=2;
        while(!(!DATA) && U8FLAG++);
        Delay_10us();
        Delay_10us();
        Delay_10us();
        U8temp=0;
        if(DATA) U8temp=1;
        U8FLAG=2;
        while((DATA) && U8FLAG++);
        //超时则跳出for循环
        if(U8FLAG==1) break;
        //判断数据位是0还是1

        // 如果高电平高过预定0高电平值则数据位为 1

        U8comdata<=<1;
        U8comdata|=U8temp; //0
    } //rof
}

```

```

uc RH(void)
{
    // 主机拉低18ms
    DATA=0;
    Delay1(180); // 原来为5
    DATA=1;
    // 总线由上拉电阻拉高 主机延时20us
    Delay_10us();
    Delay_10us();
    Delay_10us();
    Delay_10us();
    // 主机设为输入 判断从机响应信号
    DATA=1;
    // 判断从机是否有低电平响应信号 如不响应则跳出，响应则向下运行
    if(!DATA) //T !
    {
        USFLAG=2;
        // 判断从机是否发出 80us 的低电平响应信号是否结束
        while((!DATA)&&USFLAG++);
        USFLAG=2;
        // 判断从机是否发出 80us 的高电平，如发出则进入数据接收状态
        while((DATA)&&USFLAG++);
        // 数据接收状态
        COM();
        USRH_data_H_temp=UScomdata;
        COM();
        USRH_data_L_temp=UScomdata;
        COM();
        UST_data_H_temp=UScomdata;
        COM();
        UST_data_L_temp=UScomdata;
        COM();
        UScheckdata_temp=UScomdata;
        DATA=1;
        // 数据校验
        UStemp=(UST_data_H_temp+UST_data_L_temp+USRH_data_H_temp+USRH_data_L_temp);
        if(UStemp==UScheckdata_temp)
        {
            USRH_data_H=USRH_data_H_temp;
            USRH_data_L=USRH_data_L_temp;
            UST_data_H=UST_data_H_temp;
            UST_data_L=UST_data_L_temp;
            UScheckdata=UScheckdata_temp;
        }
        return 1;
    }
    else // 传感器不响应
    {
        return 0;
    }
}

```

humidity = USRH_data_H; //读出湿度，只取整数部分

电机正反转及当前模式显示函数：

```

int work(int s)//工作函数
{
    int w1=2;
    int w2=3;
    if(change==0)//判断切换是否按下
    {
        buzz(0);
        delay(20);////////////////////////
        if(change==0)
            change_flag=~change_flag;//手动，自动切换

        led_zhishi=change_flag;
        if(change_flag==1)
        {
            buzz(1);
            while(!change)//释放
            {buzz(1);}
        }

        if(change_flag==0)//自动模式
        {
            write_com(0xC6);
            write_data('A');
            delay5ms();
            if(~light==0||flag_HH==1)//夜间（无光）时拉回衣架      flag_HH==1湿度高
            {
                else if(~light==1&&flag_HH==0)//白天（有光）时伸出衣架
                {
                }
            }
            else if(change_flag==1)//手动模式
            {
                write_com(0xC6);
                write_data('M');
                if((in_forward==0)&&(limit_forward==1))//关闭是否按下
                {
                    else if((in_backward==0)&&(limit_backward==1))//打开是否按下
                    {
                        if((limit_forward==0)&&(flag_shan1==0))
                        {
                            if((limit_backward==0)&&(flag_shan2==0))
                            {
                            }
                        }
                    }
                }
            }
            return s;
        }
    }
}

```

主函数：

```

void main()//主函数
{
    uc testnum,shiwei,gewei;
    ui humidity;
    int s=6;//避免电机突然反向电压不稳的保护措施，每个方向对应一种状态，检测到状态改变先让电机停止转动再反向
    int i;
    buzz(1);
    led_zhishi=0;
    Timer0_Init();
    Init();
    init_lcd1();
    testnum = RH();
    FlagStartRH = 0;
    humidity = U0RH_data_H; //读出湿度，只取整数部分
    display();
    SendData(humidity);
    i=1;
    while(1)//进入死循环
    {
        s=work(s);//调用工作函数
        if (FlagStartRH == 1) //温湿度转换标志检查
        {
            write_com(0x8A);
            testnum = RH();
            FlagStartRH = 0;
            humidity = U0RH_data_H; //读出湿度，只取整数部分
            gewei=humidity%10+'0';
            shiwei=humidity/10+'0';
            write_data(shiwei);
            write_com(0x8B);
            write_data(gewei);
            delay5ms();
            if(humidity>=80)
            flag_HH=1;
            else
            flag_HH=0;
            TR0 = 1;
        }
    }
}

```

各类定时器及延迟函数

```

void Timer0_Init()
{
    ET0 = 1; //允许定时器0中断
    TMOD = 0x21; //定时器工作方式选择
    TL0 = 0xb0;
    TH0 = 0x3c; //定时器赋予初值
    TR0 = 1; //启动定时器
    EA=1;
}

void Init()
{
    //定时器1初始化
    TMOD=0x21;//M1:M0=10 定时器0工作方式2，可重置8位

    IT1=0;//低电平
    ET1=0;//中断允许
    TR1=1;//开启定时器计数

    TH1=0xfd;
    TL1=0xfd;//设置定时器初值
    // TH1=0xff;
    // TL1=0xe8;
    PCON=0x00;//SCON=0

    //串口初始化
    SM0=0;
    SM1=1;//串口工作方式1 8位UART
    REN=1;//使能串行接收
    ES=1;//串口中断允许
    EA=1;//开启总中断允许位
}

```

```
void Timer0_ISR (void) interrupt 1 using 0
{
    ui RHCounter;
    TLO = 0xb0;
    TH0 = 0x3c;    //定时器赋予初值

    //每5秒钟启动一次温湿度转换
    RHCounter ++;
    if (RHCounter >= 20)
    {
        FlagStartRH = 1;
        RHCounter = 0;
        TR0=0;
    }
}
```

```
void delay(ui x)//延时函数
{
    ui i,j;
    for(i=0;i<x;i++)
        for(j=0;j<110;j++);
}
```