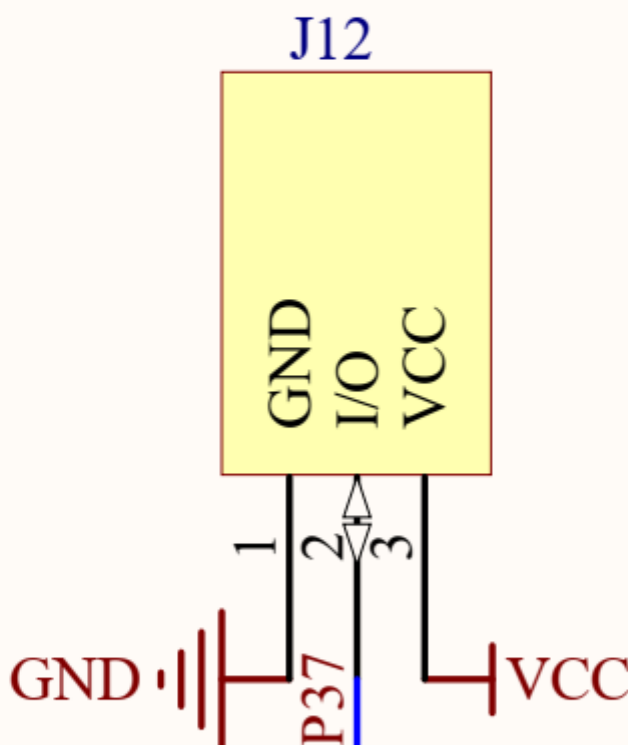


源码在这里，需要自取：

我们先来了解DS18B20

DS 18B20



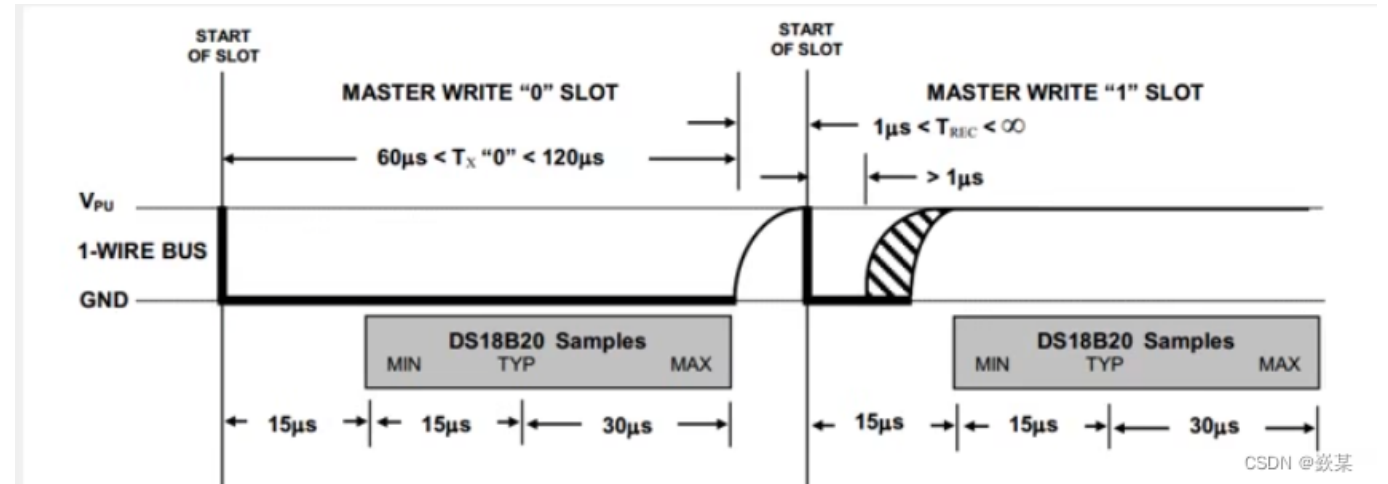
编辑

1 / 8

中间的接口是传输数据的线。****单总线，异步，半双工，**单总线只需要一根通信线即可实现数据的双向传输，当采用寄生供电（由通信线提供电压，主机配一个强上拉电路）时还可以省去VDD线路，此时，供电加通信只需要两根线。（51单片机不采取这种做法）**

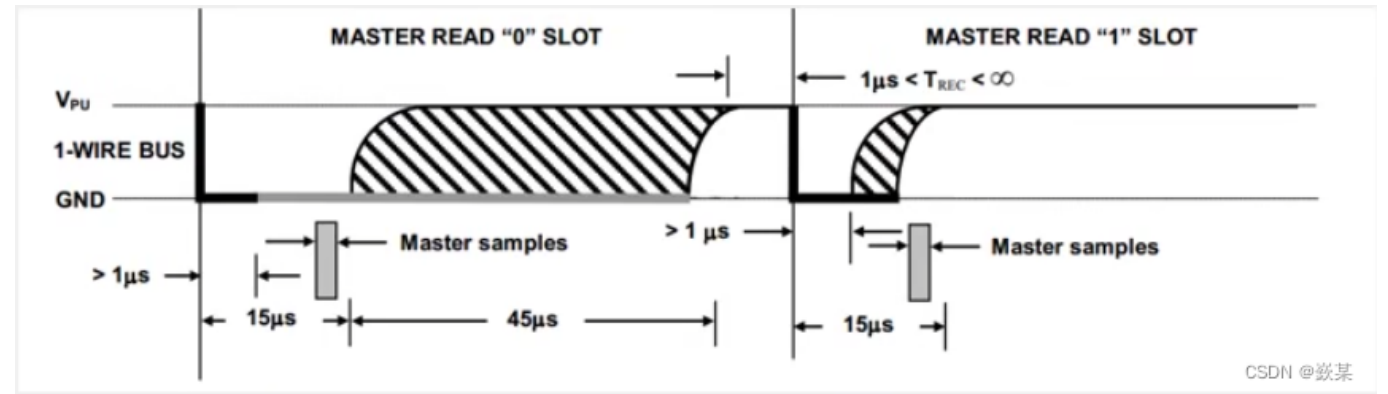
初始化：主机将总线拉低至少480us然后释放总线，等待15~60us后，从机会拉低总线60~240us响应主机，之后从机释放总线。

发送一位：主机将总线拉低60~120us，然后释放总线，表示发送0然后释放总线，表示发送1。从机将在总主机将总线拉低1~15us，（典型值）线拉低30us后读取电平，整个时间片应大于60us



编辑

接收一位：主机将总线拉低1~15us，然后释放总线，并在拉低后15us内读取总线电平(尽量贴近15us的末尾)，读取为低电平则为接收0，读取为高电平则为接收1，整个时间片应大于60us



编辑

之后就将上面两个步骤走8次，就可以封装出两个函数：发送一个字节和接受一个字节。

```
#include <REGX52.H>

//引脚定义
sbit OneWire_DQ=P3^7;

/**
 * @brief 单总线初始化
 * @param 无
 * @retval 从机响应位，0为响应，1为未响应
 */
```

```
unsigned char OneWire_Init(void)
{
    unsigned char i;
    unsigned char AckBit;
    EA=0;
    OneWire_DQ=1;
    OneWire_DQ=0;
    i = 247;while (--i);        //Delay 500us
    OneWire_DQ=1;
    i = 32;while (--i);        //Delay 70us
    AckBit=OneWire_DQ;
    i = 247;while (--i);        //Delay 500us
    EA=1;
    return AckBit;
}

/**
 * @brief 单总线发送一位
 * @param Bit 要发送的位
 * @retval 无
 */
void OneWire_SendBit(unsigned char Bit)
{
    unsigned char i;
    EA=0;
    OneWire_DQ=0;
    i = 4;while (--i);        //Delay 10us
    OneWire_DQ=Bit;
    i = 24;while (--i);        //Delay 50us
    OneWire_DQ=1;
    EA=1;
}

/**
 * @brief 单总线接收一位
 * @param 无
 * @retval 读取的位
 */
unsigned char OneWire_ReceiveBit(void)
{
    unsigned char i;
    unsigned char Bit;
    EA=0;
    OneWire_DQ=0;
    i = 2;while (--i);        //Delay 5us
    OneWire_DQ=1;
    i = 2;while (--i);        //Delay 5us
    Bit=OneWire_DQ;
    i = 24;while (--i);        //Delay 50us
    EA=1;
    return Bit;
}

/**
```

```

* @brief 单总线发送一个字节
* @param Byte 要发送的字节
* @retval 无
*/
void OneWire_SendByte(unsigned char Byte)
{
    unsigned char i;
    for(i=0;i<8;i++)
    {
        OneWire_SendBit(Byte&(0x01<<i));
    }
}

/**
* @brief 单总线接收一个字节
* @param 无
* @retval 接收的一个字节
*/
unsigned char OneWire_ReceiveByte(void)
{
    unsigned char i;
    unsigned char Byte=0x00;
    for(i=0;i<8;i++)
    {
        if(OneWire_ReceiveBit()){Byte|=(0x01<<i);}
    }
    return Byte;
}

```

之后就是在main函数里面调用这两，然后将其显示在LCD1602上。

之后的任务就是设定温度报警阈值。

定义两个变量THigh, TLow, 分别初始化为0°和30°。然后调用独立按键函数，按下Key1, THigh++; 按下Key2, THigh--; Key3, 4同理控制TLow。

```

#include <REGX52.H>
#include "Key.h"
#include "Delay.h"
/**
* @brief 获取独立按键键码
* @param 无
* @retval 按下按键的键码，范围 0~4，无按键按下，返回0
*/
unsigned char Key()
{
    unsigned char KeyNumber = 0;

```

```
    if(P3_1==0){Delay(20);while(P3_1==0);Delay(20);KeyNumber=1;}
    if(P3_0==0){Delay(20);while(P3_0==0);Delay(20);KeyNumber=2;}
    if(P3_2==0){Delay(20);while(P3_2==0);Delay(20);KeyNumber=3;}
    if(P3_3==0){Delay(20);while(P3_3==0);Delay(20);KeyNumber=4;}

    return KeyNumber;
}
```

这里将逻辑写好后会发现一个问题：

在你将按键按下后，它就不会再读取数据了。这是因为程序卡在了while循环，在你没松开之前是不会出来的。我们需要**重新封装Key.c**。这时我们就需要定时器每隔20ms调用一次它。这样既不会干扰主程序的执行，可以实现按键调节温度阈值的功能。

```
#include <REGX52.H>
#include "Delay.h"

unsigned char Key_KeyNumber;

/**
 * @brief 获取按键键码
 * @param 无
 * @retval 按下按键的键码，范围：0,1~4,0表示无按键按下
 */
unsigned char Key(void)
{
    unsigned char Temp=0;
    Temp=Key_KeyNumber;
    Key_KeyNumber=0;
    return Temp;
}

/**
 * @brief 获取当前按键的状态，无消抖及松手检测
 * @param 无
 * @retval 按下按键的键码，范围：0,1~4,0表示无按键按下
 */
unsigned char Key_GetState()
{
    unsigned char KeyNumber=0;

    if(P3_1==0){KeyNumber=1;}
    if(P3_0==0){KeyNumber=2;}
    if(P3_2==0){KeyNumber=3;}
    if(P3_3==0){KeyNumber=4;}

    return KeyNumber;
}
```

```
/**
 * @brief 按键驱动函数，在中断中调用
 * @param 无
 * @retval 无
 */
void Key_Loop(void)
{
    static unsigned char NowState, LastState;
    LastState=NowState;           //按键状态更新
    NowState=Key_GetState();      //获取当前按键状态
    //如果上个时间点按键按下，这个时间点未按下，则是松手瞬间，以此避免消抖和松手检测
    if(LastState==1 && NowState==0)
    {
        Key_KeyNumber=1;
    }
    if(LastState==2 && NowState==0)
    {
        Key_KeyNumber=2;
    }
    if(LastState==3 && NowState==0)
    {
        Key_KeyNumber=3;
    }
    if(LastState==4 && NowState==0)
    {
        Key_KeyNumber=4;
    }
}
```

然后加上蜂鸣器的报警功能，这个很简单，当 $T < T_{Low}$ 调用Buzzer，当 $T > T_{High}$ 调用Buzzer。

到这里我们的main是这样的：

```
#include <REGX52.H>
#include "LCD1602.h"
#include "DS18B20.h"
#include "Delay.h"
#include "AT24C02.h"
#include "Key.h"
#include "Timer0.h"
#include "Buzzer.h"

float T, TShow;
char TLow, THigh;
unsigned char KeyNum;

void main()
```

```
{
    DS18B20_ConvertT();    //上电先转换一次温度，防止第一次读数据错误
    Delay(1000);           //等待转换完成
    THigh=AT24C02_ReadByte(0); //读取温度阈值数据
    TLow=AT24C02_ReadByte(1);
    if(THigh>125 || TLow<-55 || THigh<=TLow)
    {
        THigh=20;           //如果阈值非法，则设为默认值
        TLow=15;
    }
    LCD_Init();
    LCD_ShowString(1,1,"T:");
    LCD_ShowString(2,1,"TH:");
    LCD_ShowString(2,9,"TL:");
    LCD_ShowSignedNum(2,4,THigh,3);
    LCD_ShowSignedNum(2,12,TLow,3);
    Timer0_Init();

    while(1)
    {
        KeyNum=Key();

        /*温度读取及显示*/
        DS18B20_ConvertT(); //转换温度
        T=DS18B20_ReadT(); //读取温度
        if(T<0)             //如果温度小于0
        {
            LCD_ShowChar(1,3,'-'); //显示负号
            TShow=-T;             //将温度变为正数
        }
        else                //如果温度大于等于0
        {
            LCD_ShowChar(1,3,'+'); //显示正号
            TShow=T;
        }
        LCD_ShowNum(1,4,TShow,3); //显示温度整数部分
        LCD_ShowChar(1,7,'. '); //显示小数点
        LCD_ShowNum(1,8,(unsigned long)(TShow*100)%100,2); //显示温度小数部分

        /*阈值判断及显示*/
        if(KeyNum)
        {
            if(KeyNum==1) //K1按键，THigh自增
            {
                THigh++;
                if(THigh>125){THigh=125;}
            }
            if(KeyNum==2) //K2按键，THigh自减
            {
                THigh--;
                if(THigh<=TLow){THigh++;}
            }
            if(KeyNum==3) //K3按键，TLow自增
            {
```

```

        TLow++;
        if(TLow>=THigh){TLow--;}
    }
    if(KeyNum==4)    //K4按键, TLow自减
    {
        TLow--;
        if(TLow<-55){TLow=-55;}
    }
    LCD_ShowSignedNum(2,4,THigh,3); //显示阈值数据
    LCD_ShowSignedNum(2,12,TLow,3);
    AT24C02_WriteByte(0,THigh);    //写入到At24C02中保存
    Delay(5);
    AT24C02_WriteByte(1,TLow);
    Delay(5);
}
if(T>THigh)    //越界判断
{
    LCD_ShowString(1,13,"OV:H");
    Buzzer_Time(50);
}
else if(T<TLow)
{
    LCD_ShowString(1,13,"OV:L");
    Buzzer_Time(50);
}
else
{
    LCD_ShowString(1,13,"    ");
}
}
}

void Timer0_Routine() interrupt 1
{
    static unsigned int T0Count;
    TL0 = 0x18;    //设置定时初值
    TH0 = 0xFC;    //设置定时初值
    T0Count++;
    if(T0Count>=20)
    {
        T0Count=0;
        Key_Loop(); //每20ms调用一次按键驱动函数
    }
}
}

```

这样我们就实现了开头视频的全部功能了。

本期博客到这里就结束了，如果有什么错误，欢迎指出，如果对你有帮助，请点个赞，谢谢！