实验二 单总线温度采集 实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| **姓名** | **班级** | **学号** | **指导老师** |
| 上官鸿 | 计算1614 | 201621121110 | 刘晋明 |

1. **实验目的**
2. 掌握数码管的使用方式
3. 掌握DS18B20温度传感器的工作原理
4. 掌握单总线通信方式实现MCU与DS18B20数据传输
5. **实验内容**
6. 学习DS18B20温度传感器的单总线传输机制，通过单片机MCU的I/O实现温度采集，并将数据显示在数码管上
7. 开启温度报警功能，设置一个阈值，利用手指触碰传感器导致温度超过阈值时，开启嗡鸣器报警
8. **实验步骤**
9. 编写数码管显示程序，下载到开发板进行调试，实现显示功能

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* DS18B20温度传感器实验 \*

实现现象：下载程序后，在温度传感器接口处，按照丝印方向插好温度传感器，数码管就会显示

检测的温度值，

注意事项：

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include "reg52.h" //此文件中定义了单片机的一些特殊功能寄存器

#include"temp.h"

**typedef** unsigned int u16**;** //对数据类型进行声明定义

**typedef** unsigned char u8**;**

sbit LSA**=**P2**^**2**;**

sbit LSB**=**P2**^**3**;**

sbit LSC**=**P2**^**4**;**

char num**=**0**;**

u8 DisplayData**[**8**];**

u8 code smgduan**[**10**]={**0x3f**,**0x06**,**0x5b**,**0x4f**,**0x66**,**0x6d**,**0x7d**,**0x07**,**0x7f**,**0x6f**};**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函 数 名 : delay

\* 函数功能 : 延时函数，i=1时，大约延时10us

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void delay**(**u16 i**)**

**{**

**while(**i**--);**

**}**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函 数 名 : datapros()

\* 函数功能 : 温度读取处理转换函数

\* 输 入 : temp

\* 输 出 : 无

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void datapros**(**int temp**)**

**{**

float tp**;**

**if(**temp **<** 0**)** //当温度值为负数

**{**

DisplayData**[**0**]** **=** 0x40**;** // -

//因为读取的温度是实际温度的补码，所以减1，再取反求出原码

temp**=**temp**-**1**;**

temp**=~**temp**;**

tp**=**temp**;**

temp**=**tp**\***0.0625**\***100**+**0.5**;**

//留两个小数点就\*100，+0.5是四舍五入，因为C语言浮点数转换为整型的时候把小数点

//后面的数自动去掉，不管是否大于0.5，而+0.5之后大于0.5的就是进1了，小于0.5的就

//算加上0.5，还是在小数点后面。

**}**

**else**

**{**

DisplayData**[**0**]** **=** 0x00**;**

tp**=**temp**;**//因为数据处理有小数点所以将温度赋给一个浮点型变量

//如果温度是正的那么，那么正数的原码就是补码它本身

temp**=**tp**\***0.0625**\***100**+**0.5**;**

//留两个小数点就\*100，+0.5是四舍五入，因为C语言浮点数转换为整型的时候把小数点

//后面的数自动去掉，不管是否大于0.5，而+0.5之后大于0.5的就是进1了，小于0.5的就

//算加上0.5，还是在小数点后面。

**}**

DisplayData**[**1**]** **=** smgduan**[**temp **/** 10000**];**

DisplayData**[**2**]** **=** smgduan**[**temp **%** 10000 **/** 1000**];**

DisplayData**[**3**]** **=** smgduan**[**temp **%** 1000 **/** 100**]** **|** 0x80**;**

DisplayData**[**4**]** **=** smgduan**[**temp **%** 100 **/** 10**];**

DisplayData**[**5**]** **=** smgduan**[**temp **%** 10**];**

**}**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名 :DigDisplay()

\* 函数功能 :数码管显示函数

\* 输入 : 无

\* 输出 : 无

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void DigDisplay**()**

**{**

u8 i**;**

**for(**i**=**0**;**i**<**6**;**i**++)**

**{**

**switch(**i**)** //位选，选择点亮的数码管，

**{**

**case(**0**):**

LSA**=**0**;**LSB**=**0**;**LSC**=**0**;** **break;**//显示第0位

**case(**1**):**

LSA**=**1**;**LSB**=**0**;**LSC**=**0**;** **break;**//显示第1位

**case(**2**):**

LSA**=**0**;**LSB**=**1**;**LSC**=**0**;** **break;**//显示第2位

**case(**3**):**

LSA**=**1**;**LSB**=**1**;**LSC**=**0**;** **break;**//显示第3位

**case(**4**):**

LSA**=**0**;**LSB**=**0**;**LSC**=**1**;** **break;**//显示第4位

**case(**5**):**

LSA**=**1**;**LSB**=**0**;**LSC**=**1**;** **break;**//显示第5位

**}**

P0**=**DisplayData**[**5**-**i**];**//发送数据

delay**(**50**);** //间隔一段时间扫描

P0**=**0x00**;**//消隐

**}**

**}**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函 数 名 : main

\* 函数功能 : 主函数

\* 输 入 : 无

\* 输 出 : 无

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void main**()**

**{**

Ds18b20SetAlarm**(**30**,** 0x00**);**

**while(**1**)**

**{**

datapros**(**Ds18b20ReadTemp**());**

DigDisplay**();**

**}**

**}**

1. 编写单总线温度采集程序

#include"temp.h"

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函 数 名 : Delay1ms

\* 函数功能 : 延时函数

\* 输 入 : 无

\* 输 出 : 无

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void Delay1ms**(**uint y**)**

**{**

uint x**;**

**for(** **;** y**>**0**;** y**--)**

**{**

**for(**x**=**110**;** x**>**0**;** x**--);**

**}**

**}**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函 数 名 : Ds18b20Init

\* 函数功能 : 初始化

\* 输 入 : 无

\* 输 出 : 初始化成功返回1，失败返回0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

uchar Ds18b20Init**()**

**{**

uchar i**;**

DSPORT **=** 0**;** //将总线拉低480us~960us

i **=** 70**;**

**while(**i**--);**//延时642us

DSPORT **=** 1**;** //然后拉高总线，如果DS18B20做出反应会将在15us~60us后总线拉低

i **=** 0**;**

**while(**DSPORT**)** //等待DS18B20拉低总线

**{**

Delay1ms**(**1**);**

i**++;**

**if(**i**>**5**)**//等待>5MS

**{**

**return** 0**;**//初始化失败

**}**

**}**

**return** 1**;**//初始化成功

**}**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函 数 名 : Ds18b20WriteByte

\* 函数功能 : 向18B20写入一个字节

\* 输 入 : 无

\* 输 出 : 无

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void Ds18b20WriteByte**(**uchar dat**)**

**{**

uint i**,** j**;**

**for(**j**=**0**;** j**<**8**;** j**++)**

**{**

DSPORT **=** 0**;** //每写入一位数据之前先把总线拉低1us

i**++;**

DSPORT **=** dat **&** 0x01**;** //然后写入一个数据，从最低位开始

i**=**6**;**

**while(**i**--);** //延时68us，持续时间最少60us

DSPORT **=** 1**;** //然后释放总线，至少1us给总线恢复时间才能接着写入第二个数值

dat **>>=** 1**;**

**}**

**}**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函 数 名 : Ds18b20ReadByte

\* 函数功能 : 读取一个字节

\* 输 入 : 无

\* 输 出 : 无

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

uchar Ds18b20ReadByte**()**

**{**

uchar byte**,** bi**;**

uint i**,** j**;**

**for(**j**=**8**;** j**>**0**;** j**--)**

**{**

DSPORT **=** 0**;**//先将总线拉低1us

i**++;**

DSPORT **=** 1**;**//然后释放总线

i**++;**

i**++;**//延时6us等待数据稳定

bi **=** DSPORT**;** //读取数据，从最低位开始读取

/\*将byte左移一位，然后与上右移7位后的bi，注意移动之后移掉那位补0。\*/

byte **=** **(**byte **>>** 1**)** **|** **(**bi **<<** 7**);**

i **=** 4**;** //读取完之后等待48us再接着读取下一个数

**while(**i**--);**

**}**

**return** byte**;**

**}**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函 数 名 : Ds18b20ChangTemp

\* 函数功能 : 让18b20开始转换温度

\* 输 入 : 无

\* 输 出 : 无

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void Ds18b20ChangTemp**()**

**{**

Ds18b20Init**();**

Delay1ms**(**1**);**

Ds18b20WriteByte**(**0xcc**);** //跳过ROM操作命令

Ds18b20WriteByte**(**0x44**);** //温度转换命令

//Delay1ms(100); //等待转换成功，而如果你是一直刷着的话，就不用这个延时了

**}**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函 数 名 : Ds18b20ReadTempCom

\* 函数功能 : 发送读取温度命令

\* 输 入 : 无

\* 输 出 : 无

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void Ds18b20ReadTempCom**()**

**{**

Ds18b20Init**();**

Delay1ms**(**1**);**

Ds18b20WriteByte**(**0xcc**);** //跳过ROM操作命令

Ds18b20WriteByte**(**0xbe**);** //发送读取温度命令

**}**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函 数 名 : Ds18b20ReadTemp

\* 函数功能 : 读取温度

\* 输 入 : 无

\* 输 出 : 无

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int Ds18b20ReadTemp**()**

**{**

int temp **=** 0**;**

uchar tmh**,** tml**;**

Ds18b20ChangTemp**();** //先写入转换命令

Ds18b20ReadTempCom**();** //然后等待转换完后发送读取温度命令

tml **=** Ds18b20ReadByte**();** //读取温度值共16位，先读低字节

tmh **=** Ds18b20ReadByte**();** //再读高字节

temp **=** tmh**;**

temp **<<=** 8**;**

temp **|=** tml**;**

**return** temp**;**

**}**

1. 开启温度报警功能，设置一个阈值，利用手指触碰传感器导致温度超过阈值时，开启嗡鸣器报警

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函 数 名 : Ds18b20ChangTemp

\* 函数功能 : 让18b20开始转换温度

\* 输 入 : 无

\* 输 出 : 无

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void Ds18b20ChangTemp**()**

**{**

Ds18b20Init**();**

Delay1ms**(**1**);**

Ds18b20WriteByte**(**0xcc**);** //跳过ROM操作命令

Ds18b20WriteByte**(**0x44**);** //温度转换命令

//Delay1ms(100); //等待转换成功，而如果你是一直刷着的话，就不用这个延时了

**if(**Ds18b20AlarmSearch**()** **!=** 0xff**)** **{** //不等于ff表示搜索有结果

BEEP **=** **~**BEEP**;**

**}**

**}**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函 数 名 : Ds18b20SetAlarm

\* 函数功能 : 设置温度告警的上阙和下阙

\* 输 入 : TH TL

\* 输 出 : 无

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void Ds18b20SetAlarm**(**int TH**,** int TL**)**

**{**

Ds18b20Init**();**

Delay1ms**(**1**);**

Ds18b20WriteByte**(**0xCC**);**

Ds18b20WriteByte**(**0x4E**);**

Ds18b20WriteByte**(**TH**);**

Ds18b20WriteByte**(**TL**);**

**}**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函 数 名 : Ds18b20AlarmSearch

\* 函数功能 : 告警搜索

\* 输 入 : 无

\* 输 出 : 搜索结果

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int Ds18b20AlarmSearch**()** **{**

Ds18b20Init**();**

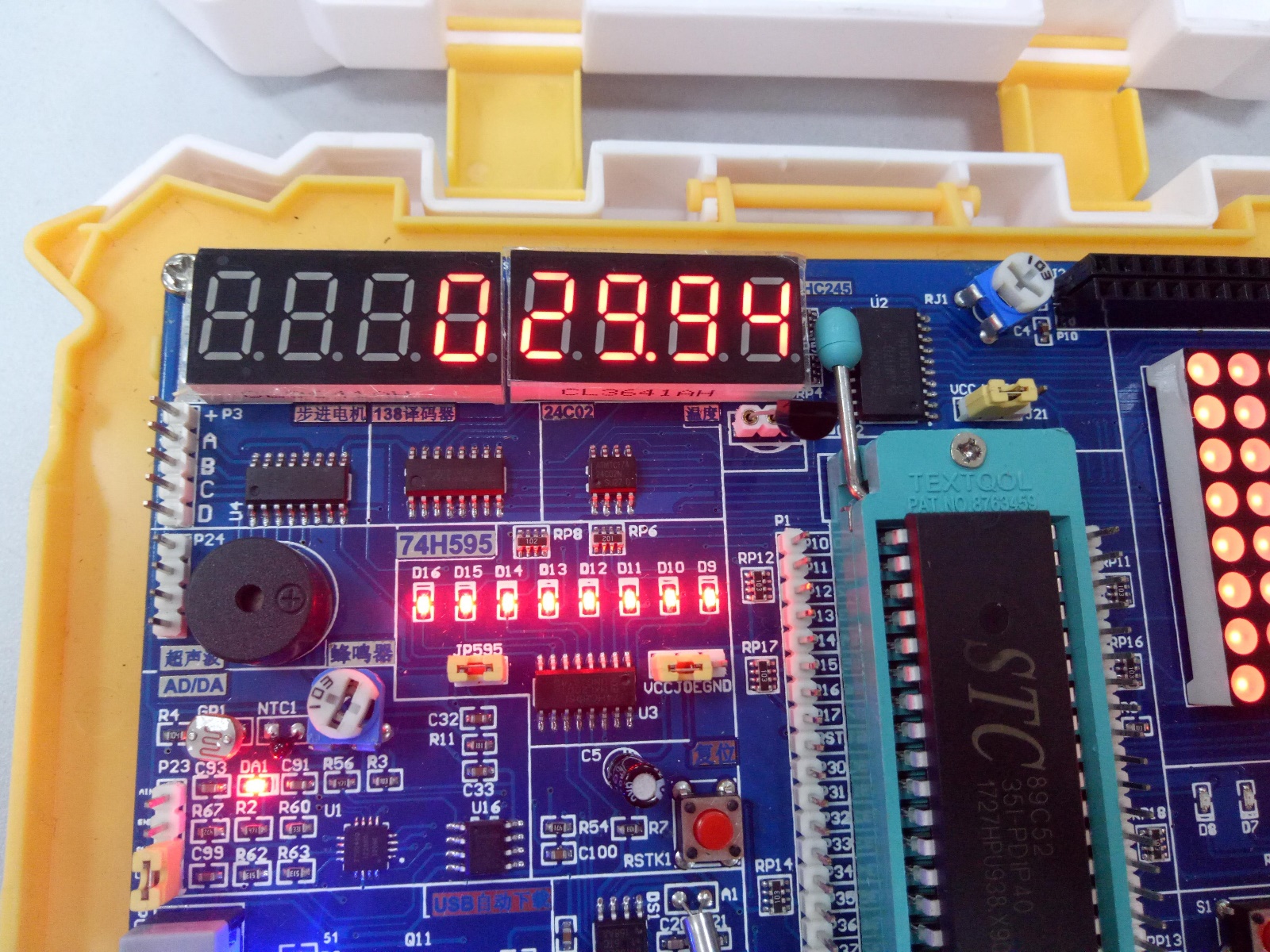
Delay1ms**(**1**);**

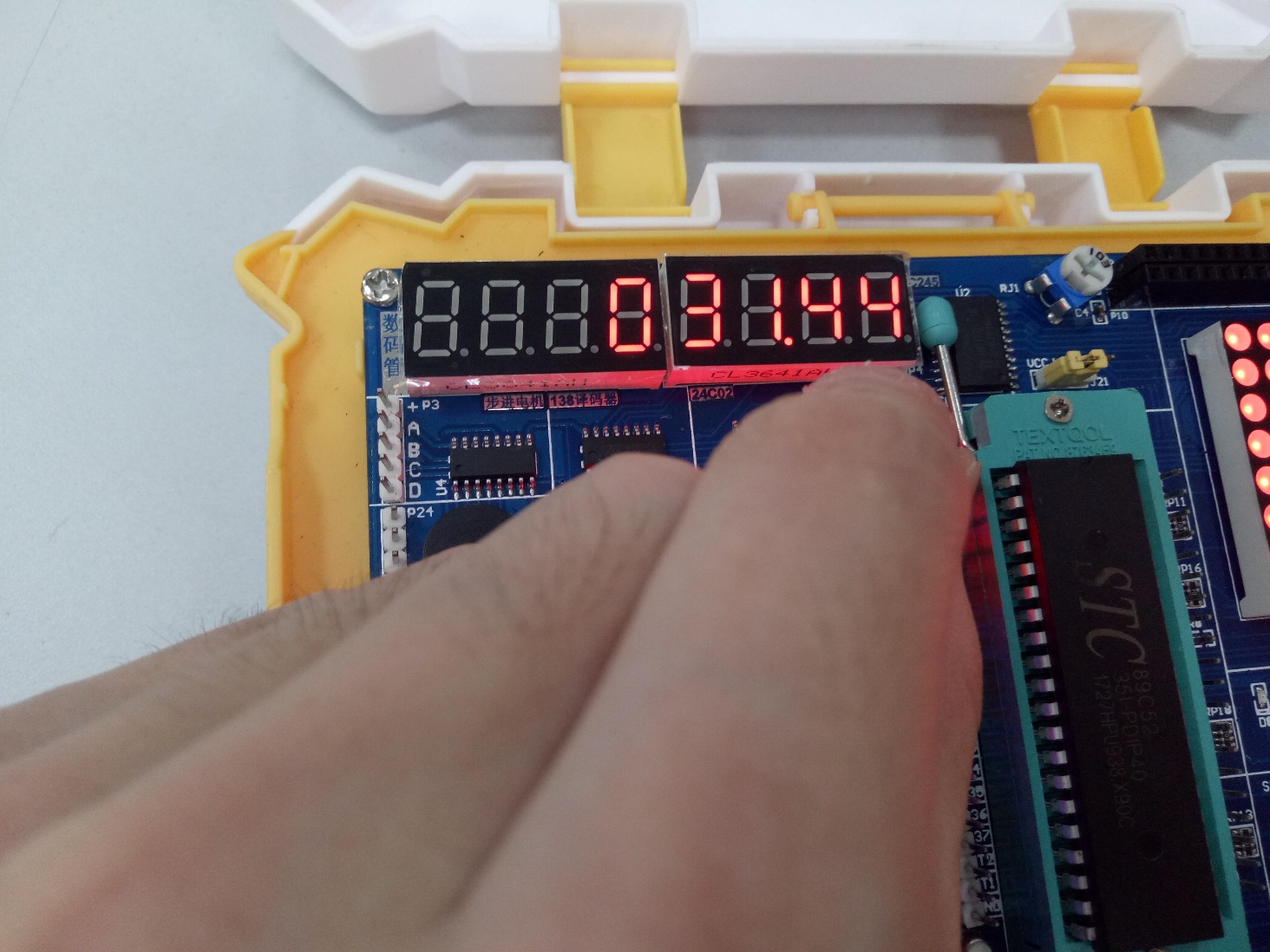
Ds18b20WriteByte**(**0xEC**);**

**return** Ds18b20ReadByte**();**

**}**

1. 实验结果







1. **实验总结**
2. 掌握了数码管的使用方式
3. 掌握了DS18B20温度传感器的工作原理
4. 掌握了单总线通信方式,实现了MCU与DS18B20数据传输
5. 能够实现当温度超过一定阙值时,启动蜂鸣器报警功能