

1. 概述

现代信息技术的三大基础是信息采集、信息传输和信息处理。传感器属于信息技术的前沿尖端产品，尤其是温度传感器被广泛用于工农业生产、科学研究和生活等领域，数量高居各种传感器之首。本系统以 MSP430 系列的单片机为核心，加入了无线传输模块与活动探头，成功实现了准确，实时，灵活地测量环境温度。通过加入按键控制逻辑与温度控制模块，本系统还可以实现多路温度巡检，报警与温度控制功能。因而具有较强的使用性。

本报告以下的内容将会按照以下结构来组织：在第二小节中我们将会介绍设计的目标；第三小节中，我们将会介绍组员分工情况；第四小节的内容是系统整体功能的介绍；第五小节里，每一个小模块的软硬件设计与性能将会被详细介绍；我们系统整机测试的过程与结果将会展现在第六小节。

2. 设计目标

1.1 基本功能

- 温度测量范围为 $0^{\circ}\text{C}\sim 45^{\circ}\text{C}$ ，分辨率为 0.1°C
- 具有温度数码显示功能，分辨率为 0.1°C 。
- 具有温度超限报警功能，当温度超出指定温度，必须给出声或光提示信号。温度报警的阈值可由按键控制。
- 进行多路温度巡检，显示当前巡检传感器的温度测量值；可通过按键选择工作在巡检或指定传感器测量模式。
- 传感器与主控制机之间通过无线收发装置进行数据通信。

1.2 高级功能

- 进行温度的自动调节控制，可调节范围为 $30^{\circ}\text{C}\sim 45^{\circ}\text{C}$ ，最小设定分度为 1°C 温度控制范围可由外界输入，当温度达到某一设定值并稳定后，装置接触表面的温度波动范围控制在 $\pm 5^{\circ}\text{C}$ 以内。温度调控达到稳定状态时，发出提示信号灯。
- 能记录并实时显示温度变化过程的曲线，显示的误差绝对值小于 2°C 。

3. 团队与分工

本小组分工的指导思想如下：为了最大程度实现并行性，我们按照模块来划分任务。不同模块之间首先需要将相互之间的接口定义好，定义完成以后不同模块就能够相对地独立工作了。我们所划分的四大模块为：传感器模块，无线收发模块，主控制机模块与温度控制模块。

黄华 负责传感器模块的电路与程序的设计与调试；协助无线模块发送端的调试。进行温度控制模块的编程。进行元器件采购。

倪鼎 负责无线传输模块的电路与程序的设计和调试；协助传感器模块与主控制机模块的调试。进行电路板的焊接。进行整机测试。

刘曦文 负责温度控制模块的设计，实现与调试。进行电路板的焊接。进行元器件采购。小组实验报告修订。

张书宇 负责 LCD 显示驱动，键盘驱动的设计与实现；完成多路巡检，声光报警，输入报警阈值温度等逻辑功能；进行整机测试。制作宣传视频。

4. 总体设计方案与应用场景。

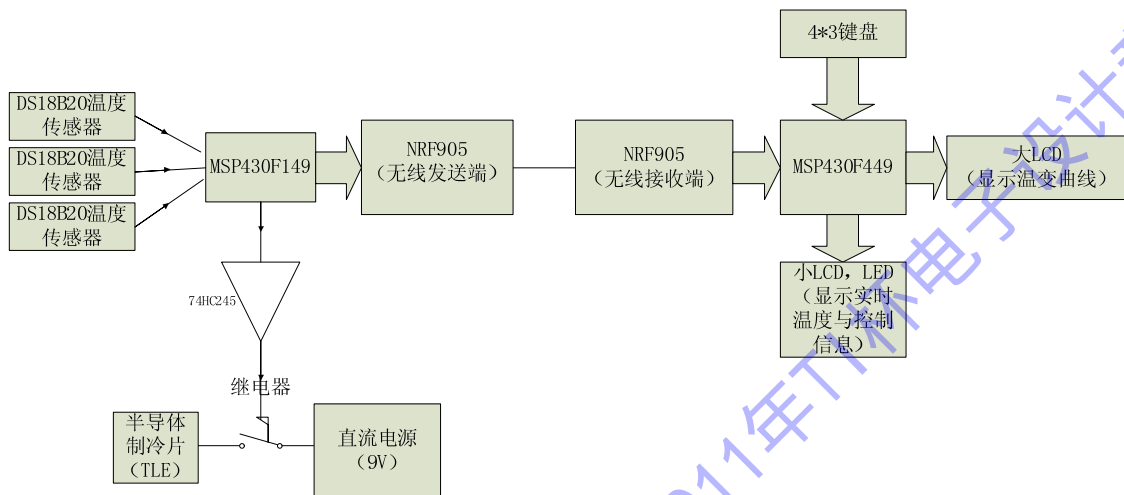


图 1. 总体设计框图

5. 软硬件设计与实现

5.1. 温度传感模块

5.1.1. 综述

本模块选用了数字温度传感器 DS18B20，它的测温范围为 $-55^{\circ}\text{C} \sim +125^{\circ}\text{C}$ ，可在 1 s (典型值) 内把温度转换成数字。最高可以实现 12 位温度转换精读，使最高温度精度达到 0.0625°C 。从这些数据可以看出，本传感器可以满足系统设计中的测量温度范围，精度与转换时间的要求。

除此以外，本传感器还具有独特的单线通信功能。仅仅需要一条线就可以将传感器与 CPU 连接起来。因而 DS18B20 同时具备有接线方便，占用端口少，外围电路简单的优点。

驱动数字传感器的 CPU 型号为 MSP430F149。它有着处理能力强，运算速度快，超低功耗，开发环境方便高效的特点。

5.1.2. 硬件电路设计

主要参数

■ 电器特性。

DS18B20 的供电电压为 $3.0\text{V} \sim 5.0\text{V}$ 。MSP430 的供电电压为 3.3V 。因而可以直接利用 MSP430 的供电电源来为 DS18B20 供电。

■ 时序特性

由于DS18B20通过单线与CPU通信，因而时序在通信正确与否方面起到了至关重要的作用。仔细分析它的时序图，可知DS18B20所要求的最短读写间隔为 $15\mu s$ ，而MSP430F149的时钟频率为 $8MHz$ ，即时钟周期为 $0.125\mu s$ 。两者相差两个数量级，因而只要通过恰当的延时，MSP430F149就能准确地和DS18B20进行数据通信。

■ 测量精度

本传感器的测量标称误差为 $\pm 0.5^{\circ}C$ ，测量数据分辨率为 $0.0625^{\circ}C$ 。温度采集时间的最大值为 $750ms$ 。基本满足本系统的精度要求。

电路连接方式

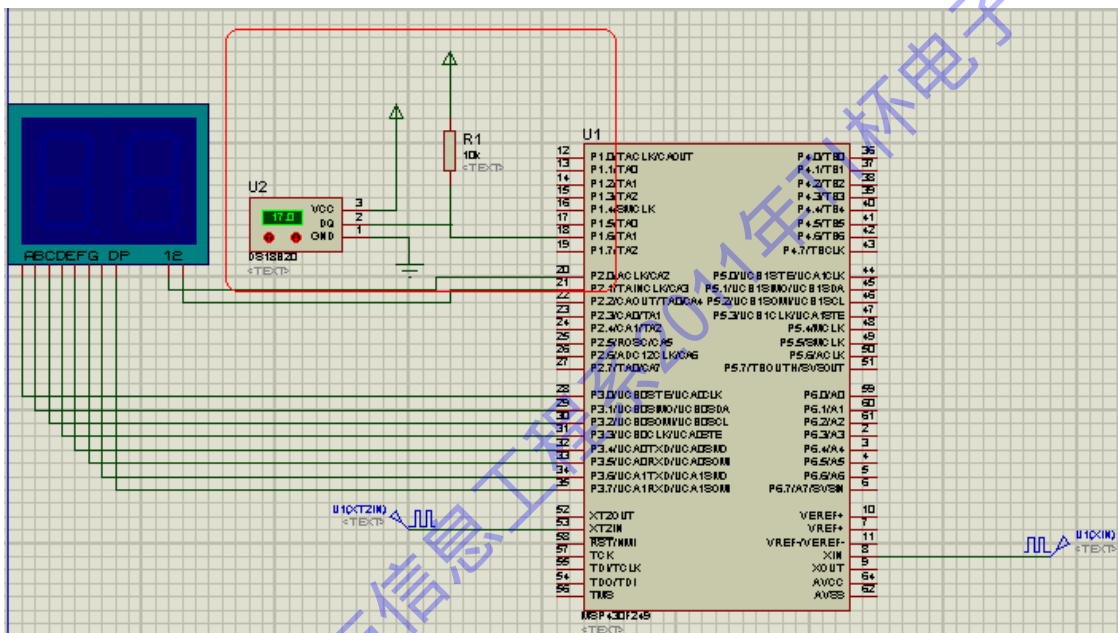


图2.

由仿真图可见，DS18B20的3号引脚接3.3V电源，2号引脚通过一个5k的电阻连接到3.3V，同时该引脚也直接连接到CPU的数据端口。1号引脚接地。

5.1.3. 采温流程

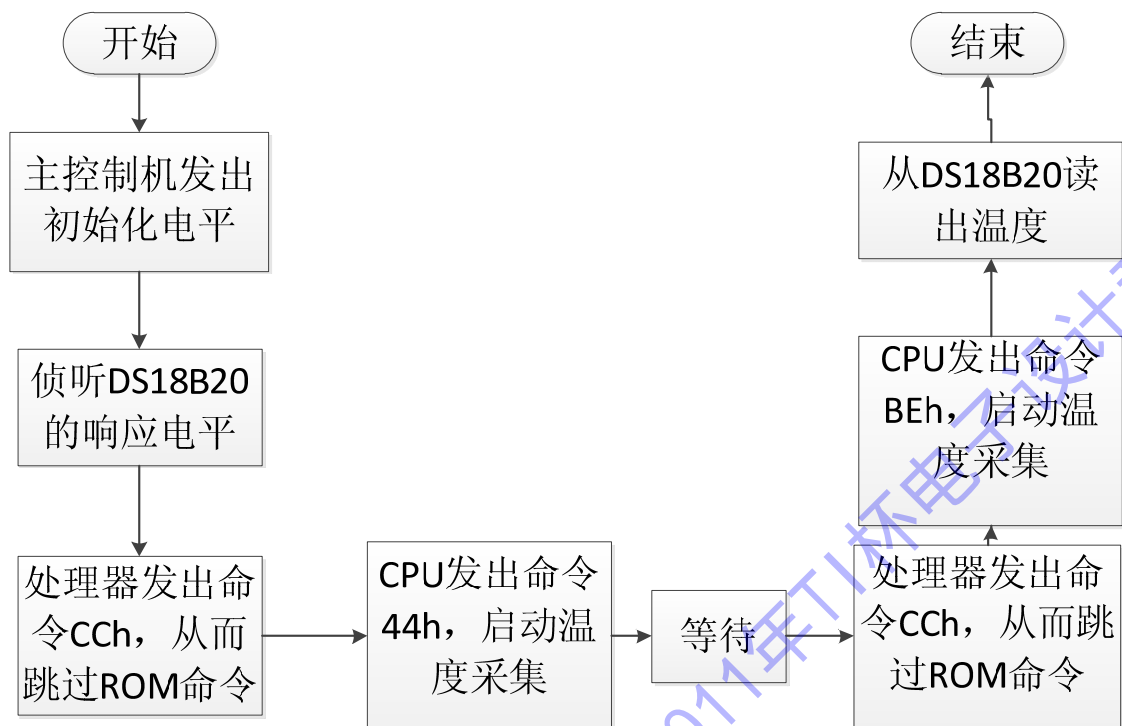


图 3

5.1.4. 软件模块

- void DS18B20_I0_set(void);

本函数实现了在MSP430的编程环境下，DS18B20的初始化。根据传感器自身的需要，本函数实现了把相应的端口设置成标准输入输出模式。

- void ds18b20_reset(void)

本函数对应于流程图中的前两步，它的作用在于唤醒传感器，使之进入工作状态。具体的实现方法是：CPU先将电平拉低，当DS18B20检测到这个电平变化后，会在15-60 μ s后回应一个低电平给CPU。通过这样一个流程，CPU与DS18B20均可以确认对方的存在。

- void ds18b20_write(char Date)

本函数对应于流程图中的处理器向传感器写命令的步骤。要实现这个功能，函数中必须严格遵守传感器的时序要求。具体时序如下：

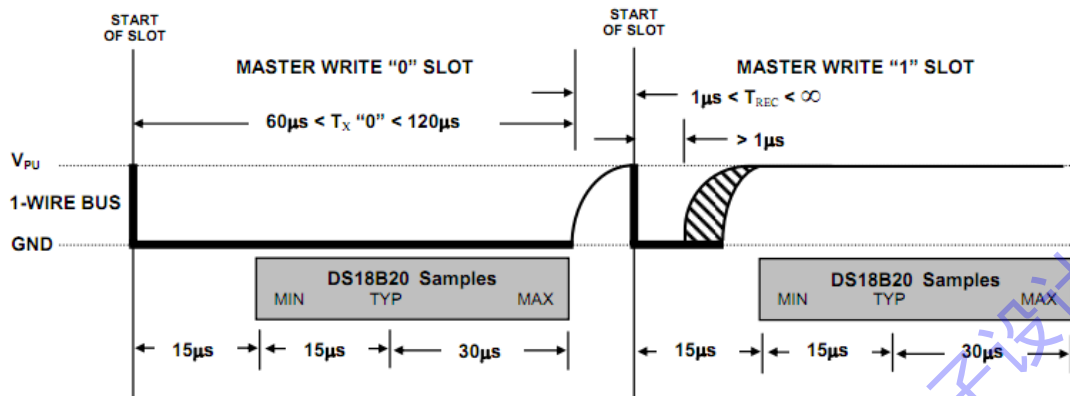


图 4

当写入一个逻辑“0”时，CPU 将电平拉低 60–120 μ s，在这段时间里面，DS18B20 会进行数据采样。在低电平结束后，电平会在外接电源的作用下恢复高电平。在这以后就可以进行下一个比特的输入。当输入逻辑“1”时，CPU 先将电平拉低超过 1 μ s，然后在 15 μ s 内转换成高电平，并且持续超过 45 μ s。这样，DS18B20 就能够正确采样到数据。

本函数就是依照这样的时序要求，将输入的数据逐字地输入给 DS18B20。

● char ds18b20_read(void)

本函数对应于流程图中的温度读取步骤。其所遵从的时序如下：

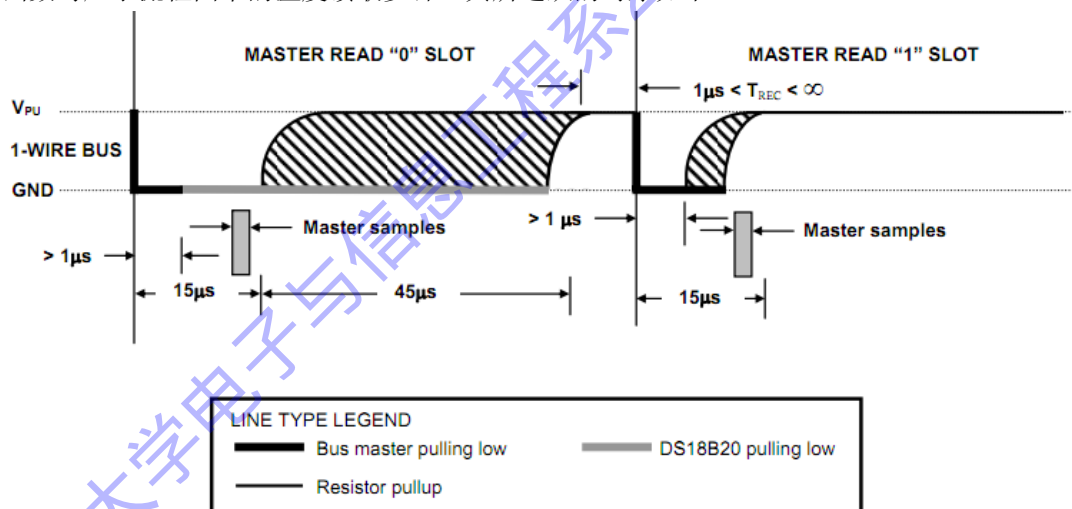


图 5

首先，CPU将电平拉低超过 1 μ s，相当于向DS18B20发出一个启动信号。如果DS18B20将要输出一个逻辑“0”，那么它将在随后至少 15 μ s 内保持低电平，CPU只需要在这段时间内读取电平，就能够读出逻辑“0”。如果DS18B20将要输出逻辑“1”，那么它将在电平在拉低后 15 μ s 之内让电平重新恢复高电平。只要CPU在 15 μ s 读取引脚电平，那么就能把逻辑“1”读出。

● void DS1820_delay(int i)

```
{while(i--);}
```

本函数实现DS18B20的读写过程中必要的延时。MSP430F149的默认时钟频率为8MHz，每条指令需要四个时钟周期。由于 $1 \div 8MHz = 0.125\mu s$ ，因而处理器执行两条指令的时间为 1 μ s。本函数

恰好执行了 $2i$ 条指令，故延迟时间为 $i\mu s$ 。

5.1.5. 仿真结果

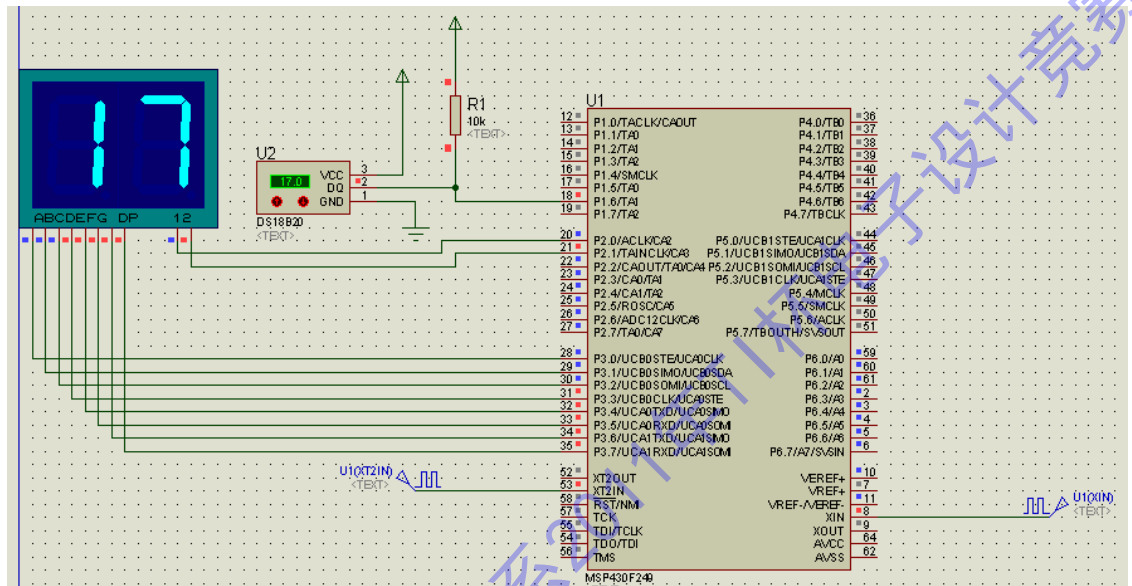


图 6

由于在Proteus7.7的软件库中没有MSP430F149这一款型号，因而我们选用了MSP430F249。两者只有细微的差别，程序也能够方便地相互移植。

5.1.6. 模块测试结果

首先，我们验证温度正确的第一种方法为比较我们所使用的三个传感器所采集的温度信息。根据观察我们可以发现，三个传感器所采集到的温度信号相差无几，一般都在 0.3°C 以内。同时考虑到这一款传感器的标称测量误差为 $\pm 0.5^{\circ}\text{C}$ 。在这样的条件下，我们所采集到的温度发生错误是小概率事件。由此，我们基本上可以断定温度采集正确。

另外，我们尝试测试传感器的温度转换速度。我们的测试方法如下：我们将LCD显示器的刷新频率调整为1秒一次，然后用手捏住传感器使之升温。这样我们就可以在每次屏幕刷新时看见温度值的变化，表明在1秒钟以内温度已经完成了转换。这也符合DS18B20说明文档所说的转换速率不低于730ms每次的指标。由于我们最终产品的屏幕刷新频率为1.3秒一次，因而传感器的转换速率能够满足系统的需求。

5.2. 无线传输模块

5.2.1. 概述

经过反复比较，我们最终选择使用Nordic公司生产的nRF905无线收发模块作为我们的通信芯片。nRF905单片无线收发器工作在433/868//9185的ISM频段。有一个完全集成的频率调

制器,一个带解调器的接收器,一个功率放大器,一个晶体振荡器和一个调节器组成。ShockBurst 工作模式的特点是自动产生前导码和 CRC。可以很容易通过 SPI 接口进行编程配置。电流消耗很低,在发射功率为-10db m 时,发射电流为 11mA,接收电流为 12.5mA,进入 powerdown 模式可以很容易实现节电。

具体参数如下:

参数	数值	单位
最低工作电压	1.9	V
最大发射功率	10	dBm
最大数据传输率(曼切斯特编码)	100	kbps
输出功率为-10 dBm 时工作电流	11	mA
接收模式时工作电流	12.5	mA
温度范围	-40 to +85	°C
典型灵敏度	-100	dBm
POWERDOWN 模式时工作电流	2.5	uA

表 1 nRF905 快速参考数据

nRF905 如图所示:



图 7 nRF905 实物图

nRF905 的内部结构图如下所示:

华中科技大学电子与信息工程系2011年电子设计竞赛

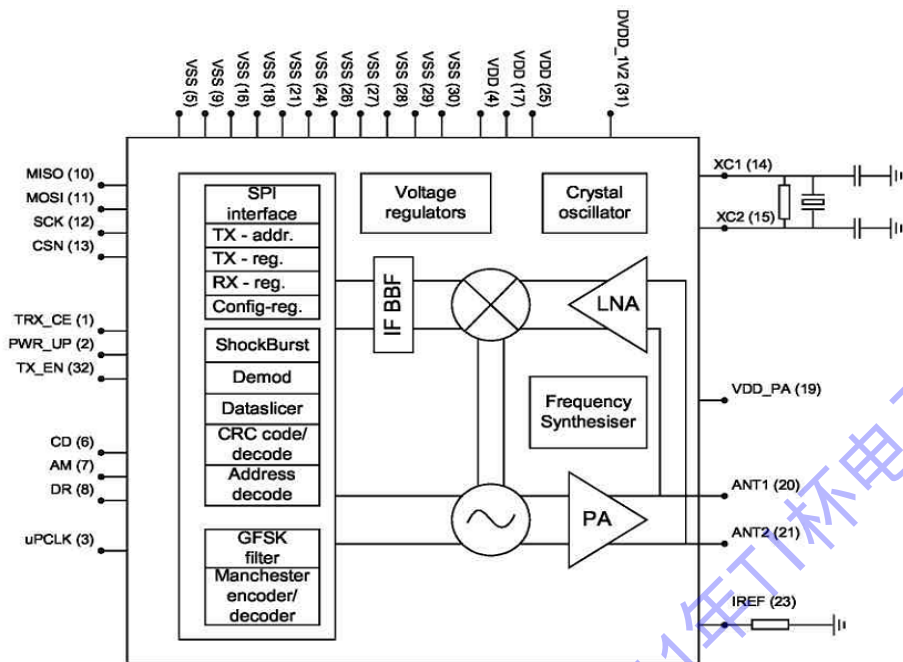


图 8 nRF905 内部结构图

5.2.2. 工作流程

SPI总线运行流程：

nRF905 的所有配置都通过 SPI 接口进行。SPI 接口由 5 个寄存器组成，一条 SPI 指令用来决定进行什么操作。SPI 接口只有在掉电模式和 Standby 模式是激活的。

- 状态寄存器 (Status-Register)
寄存器包含数据就绪 DR 和地址匹配 AM 状态。
- RF 配置寄存器 (RF-Configuration Register)
寄存器包含收发器的频率、输出功率等配置信息。
- 发送地址 (TX-Address)
寄存器包含目标器件地址，字节长度由配置寄存器设置。
- 发送有效数据 (TX-Payload)
寄存器包含发送的有效 ShockBurst 数据包数据，字节长度由配置寄存器设置。
- 接收有效数据 (RX-Payload)
寄存器包含接收到的有效 ShockBurst 数据包数据，字节长度由配置寄存器设置。在寄存器中的有效数据由数据准备就绪 DR 指示。

发送流程：

SPI 接口由 CSN、SCK、MOSI 以及 MISO 组成。在配置模式下单片机通过 SPI 接口配置高频头的工作参数；在发射/接收模式下单片机 SPI 接口发送和接收数据。整个 nRF905 的操作都是在 SPI 的控制下，通过操作命令字操作内部的相应寄存器，实现芯片的工作模式的配置和数据的收发。在收发模式下，nRF905 的工作流程如下：

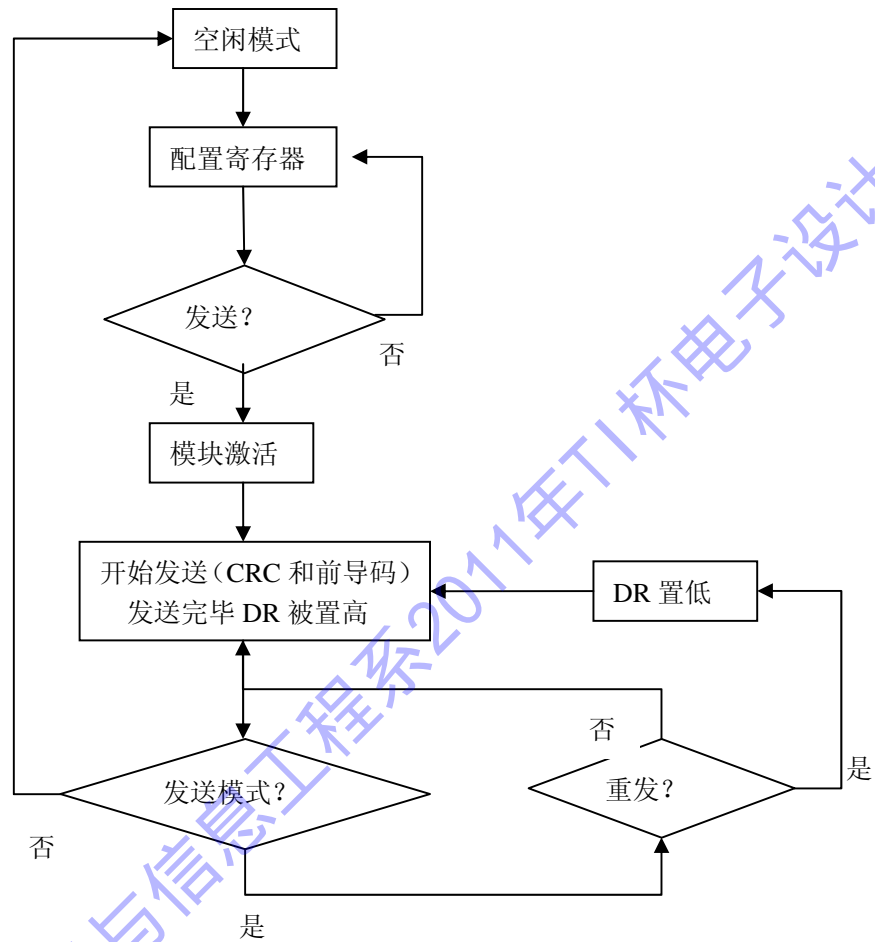


图9 nRF905发送流程图

- 1) 当应用 MCU 有遥控数据节点时，接收节点的地址 TX-address 和有效数据 TX-payload 通过 SPI 接口传送给 nRF905 应用协议或 MCU 设置接口速度；
- 2) MCU 设置 TRX_CE、TX_EN 为高来激活 nRF905 ShockBurst 传输；
- 3) nRF905 ShockBurst：
 - 无线系统自动上电
 - 数据包完成（加前导码和 CRC 校验码）
 - 数据包发送（100kbps，GFSK，曼彻斯特编码）
- 4) 如果 AUTO_RETRAN 被设置为高 nRF905 将连续地发送数据包直到 TRX_CE 被设置为低；
- 5) 当 TRX_CE 被设置为低时，nRF905 结束数据传输并自动进入 standby 模式。

接收流程：

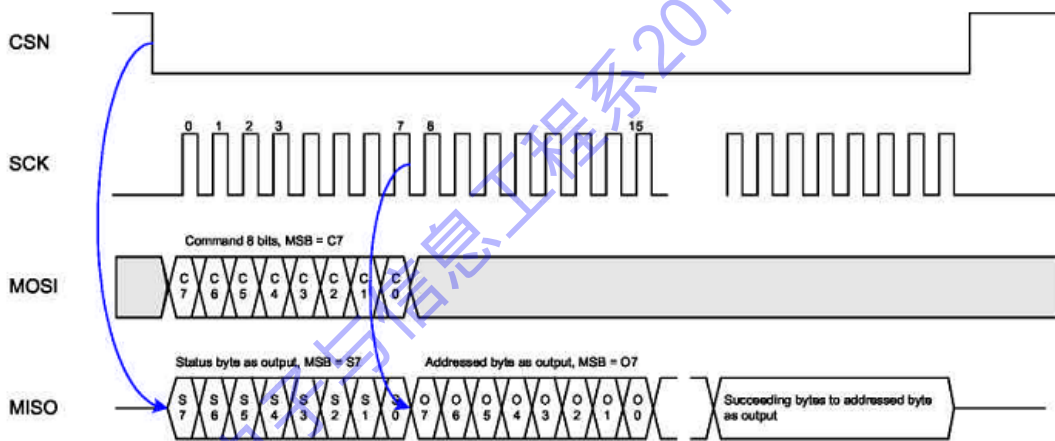
- 1) 通过设置 TRX_CE 高，TX_EN 低来选择 ShockBurst 模式；
- 2) 650us 以后，nRF905 监测空中的信息；

- 3) 当 nRF905 发现和接收频率相同的载波时，载波检测 CD 被置高；
- 4) 当 nRF905 接收到有效的地址时，地址匹配 AM 被置高；
- 5) 当 nRF905 接收到有效的数据包（CRC 校验正确）时，nRF905 去掉前导码、地址和 CRC 位，数据准备就绪（DR）被置高；
- 6) MCU 设置 TRX_CE 低，进入 standby 模式低电流模式；
- 7) MCU 可以以合适的速率通过 SPI 接口读出有效数据；
- 8) 当所有的有效数据被读出后，nRF905 将 AM 和 DR 置低；
- 9) nRF905 将准备进入 ShockBurst RX、ShockBurst TX 或 Powerdown 模式。

5.2.3. 工作时序

nRF905 使用 SPI 接口实现命令控制和数据传递。SPI 接口由 CSN、SCK、MOSI 以及 MISO 组成。每条命令的执行都是在 CSN 片选信号的下降沿开始，在 CSN 为低电平时段执行指令，依次每条指令的执行都要先将 CSN 置低电平，指令执行结束后立即将 CSN 电平置高。在 CSN 为低电平的有效时段，在 SCK 的电平脉冲的下降沿读写 1bit 的数据，因此每条指令或数据的读写操作都应该有 8 次或 8 的周期倍的 SCK 的电平切换。读写的工作时序如下所示：

SPI 时序



在读操作时，在 CSN 片选有效和 SCK 脉冲的同步下，数据从 nRF905 的接收数据寄存器中按字节以 Bit 为的方式从 MISO 口输出；在写操作时，在 CSN 片选有效和 SCK 脉冲的同步下，待发送的数据从 MCU 按字节以 Bit 为的方式从 MOSI 口输入。在源代码中，通过函数 `unsigned char SpiRead(void)` 实现。

5.2.4. 电路设计

在收发端，将 nRF905 的控制端口、串行数据端口和脉冲同步端口分别与 MCU 的通用 I/O 口相连，通过 MCU 的通用 I/O 口的电平状态的变化实现与 nRF905 的数据交互。例如，这里我们将 nRF05 的 SCK 信号线接到 MSP430 的通用 P2.4 (P6.4) 口上，通过控制 MSP430 的 I/O 口的输出电平来产生读写时序脉冲，这样就方便的解决了 MSP430 与 nRF905 读写时的时钟同步问题，读写速

率由 MSP430 的 I/O 口的电平切换速率决定。这样整个读写过程简单可控。

基于上述 nRF905 的工作原理，设计的无线收发模式的硬件连线图如下所示：

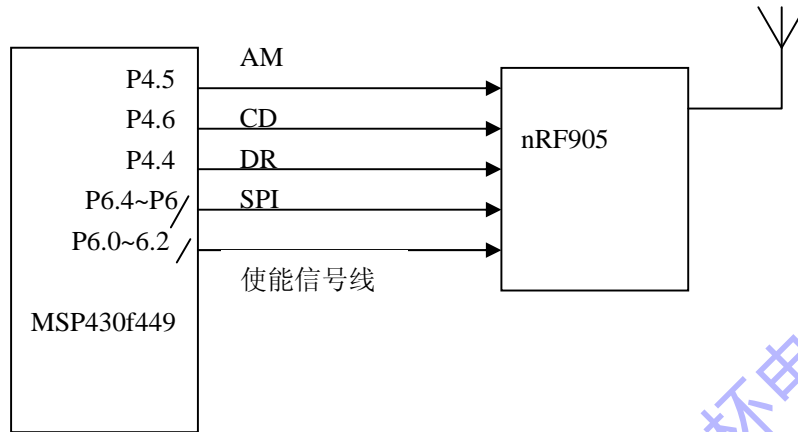


图 10. 接收端

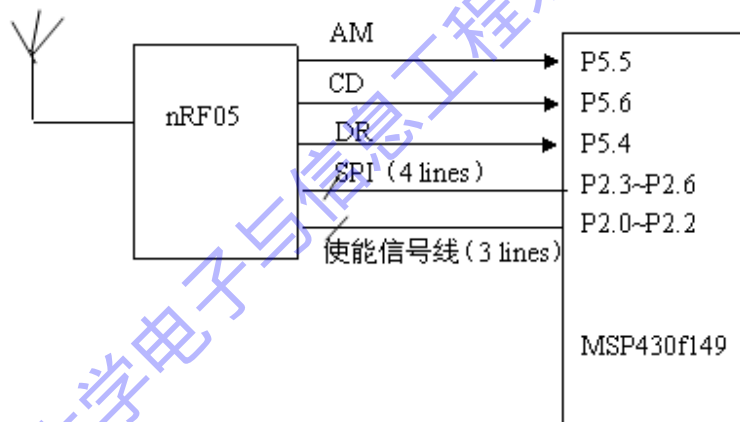


图 11 发送端

按照 nRF905 的设计特点，启动无线工作模块时，首先要配置 RF 寄存器，其中包括发（接收）频率，发送（接收）功率，地址宽度，发送（接收）有效字节长度，接收地址，校验，晶振频率等。另外，nRF905 使用的是单线多控的 SPI 工作方式，所有操作都是由预定义的操作命令指定，在每一次操作前，都要先写操作命令码，紧接着才是相应的操作指令。同样，任何操作都受 SCN 的控制。基于上述特点，nRF905 的工作流程图如下所示：

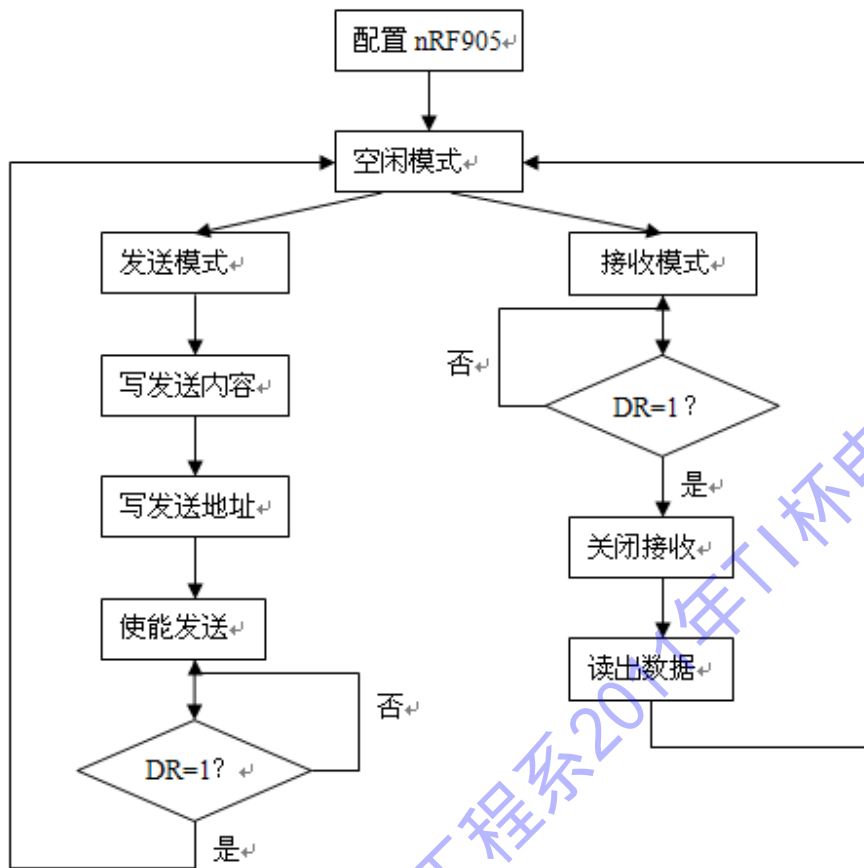


图 12 nRF905 的工作流程图

5.2.5. 程序实现

发送数据包程序:

```

void TxPacket(void)
{
    uchar i;
    CSN_0;           // 打开SPI
    SpiWrite(WTP);  // 待发数据装载命令
    for (i=0;i<32;i++)
    {
        SpiWrite(TxRxBuf[i]); //写入待发送数据
    }
    CSN_1;           // 关闭SPI
    Delay(1);
    CSN_0;           // 打开SPI
}
  
```

```

SpiWrite(WTA);          // “写发送地址” 命令
for (i=0;i<4;i++)      // 4字节地址
{
    SpiWrite(TxAddress[i]);
}
CSN_1;                 //关闭SPI
TRX_CE_1;             // Set TRX_CE high, start Tx data transmission
Delay(10);            // while (DR!=1);
TRX_CE_0;             // Set TRX_CE low
}

```

接收数据包程序:

```

void RxPacket(void)
{
    char i;
    Delay(100);
    TRX_CE_0;
    CSN_0;              // SPI使能 (Spi enable for write a spi command)
    Delay(1);
    SpiWrite(RRP);     // 读SPI数据命令 (Read payload command)
    for (i = 0 ;i < 32 ;i++)
    {
        TxRxBuf[i]=SpiRead();// Read data and save to buffer
    }
    CSN_1;             // 关闭SPI
    Delay(10);
    TRX_CE_1;
}

```

通过以上的发送与接收程序可以实现数据的发送和接收。所有的数据都存储在 32 字节的 TxRxBuf(32) 数组中, 上(下)位机只要到该数组中提取或赋值就可以实现数据的无线收发。这样就方便的解决了无线收发模块与 MCU 的数据接口, 操作起来十分简单。

5.3. 上位机驱动模块

5.3.1. 概述

本项目中, 我们上位机的选择是利尔达 MSP430F449TEST44X 实验箱为平台, 优点如下:

- 节省经费更多利用在无线与温控模块;
- 实验箱有大量可行代码供借鉴, 减小了基础的开发难度;
- 实验箱有较完整的硬件连接, 节约大量的制版时间。

考虑到上位机是对多模块联合调试的中枢以及先决条件, 同时其预期的显示功能对调试提供

了重要支持，在开发进度上必须适当超前。因而，我们选用了具有上述优势的实验箱方案。

5.3.2. 设计目标

- 实时测温功能：接受无线发送端数据，解析、转存并用数字显示。由于探头输出的数值精度达十六分之一（约 0.06°C ），而探头额定误差为 0.5°C ，故最终显示数据保留小数点后 1 位。
- 警报功能：当所测实时温度超出预设阈值时，发出蜂鸣器警报，并辅以 LED 显示；温度的阈值可由键盘输入改变。
- 路选/巡检功能：路选——可固定选择 1 到 3 路中的某一路温度输出；巡检——每隔约 1S 按 1→2→3→1 的顺序换路；换路的同时，所有外设实时同步输出。
- 记录曲线绘制功能：记录收到的 100 个数据，绘制曲线；当收到新数据时，更新曲线，表现温度的变化趋势；由按键输入，可以放缩曲线的尺寸，当温度缓慢变化时可显示温度的细微变化，当温度大范围变化时，能整体显示曲线；可上下平移曲线并显示中线（横轴）所示温度，便于将曲线显示于屏幕中央，并可粗略测量记录中任意时刻的温度（精度 1 摄氏度）。

5.3.3. 硬件设计

- 键盘

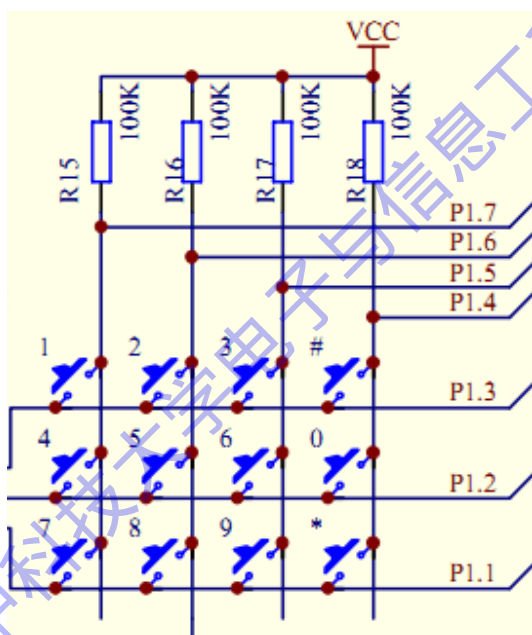


图 13

键盘采用 3 行线 4 列线 7 线制，使用端口 P1.1-P1.7。经测试，电路硬件防抖效果不佳，故采用轮询方式读出键值，并通过软件延时去抖。

- LED 数码管

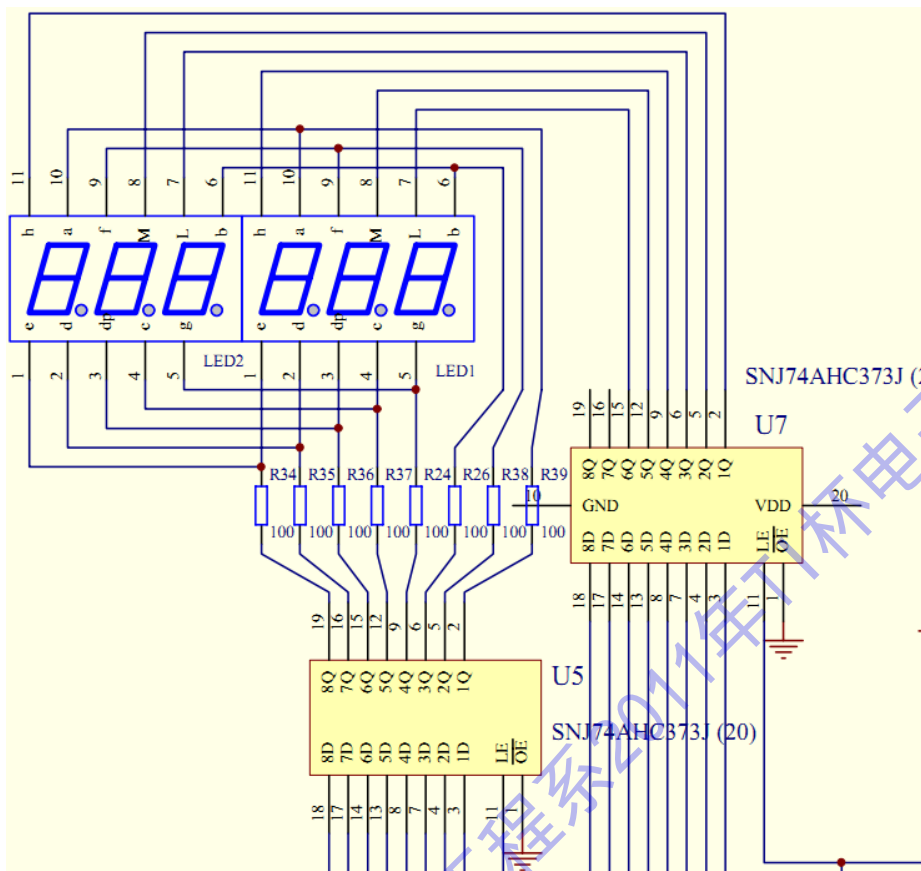


图 14

模块总线如下：

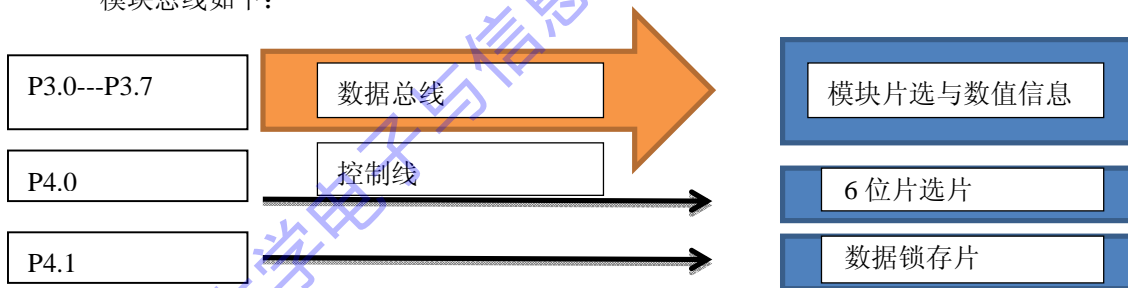


图 16

通过 P4.0 使能片选，P4.1 输出值，使 6 位 LED 分时工作。因此，为保持 LED 的显示效果，应做出适当时延，并定时刷新。

- LCD 点阵

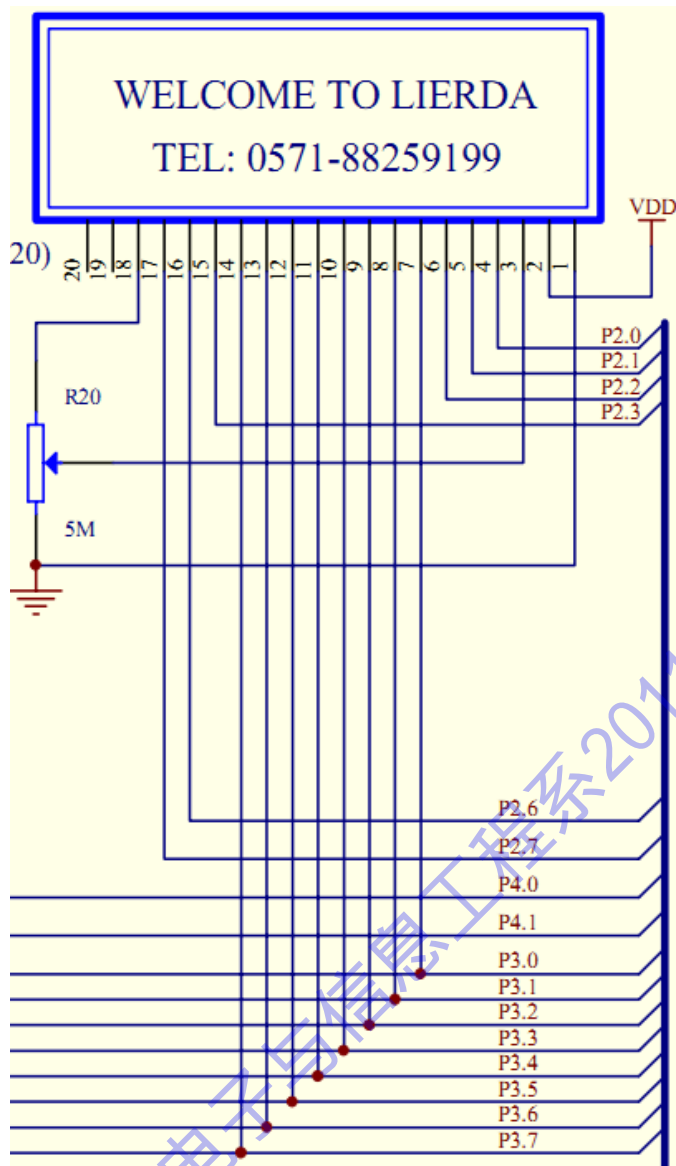


图 17

- P2. 0: RS 端
- P2. 1: RW 端
- P2. 2: EN 端
- P2. 3: CS1 低 64 位使能端
- P2. 6: CS2 高 64 位使能端
- P2. 7: RST 端

根据实际输出的流程，先选择 0-7 行中的一行使能，再选择 0-127（分为两部分使能）列中的一行使能，使能一行 8 点，从上至下为低位到高位，由数据线写入数据。

12864LCD 模块设计中两个重点：

- 1) LCD 具有记忆效应，故刷新图像必须将前一次的图像清零；
- 2) LCD 与 LED 共享数据总线 P3，故设计时应注意数据保护。

- 数码 LCD 模块

模块电路图如下：

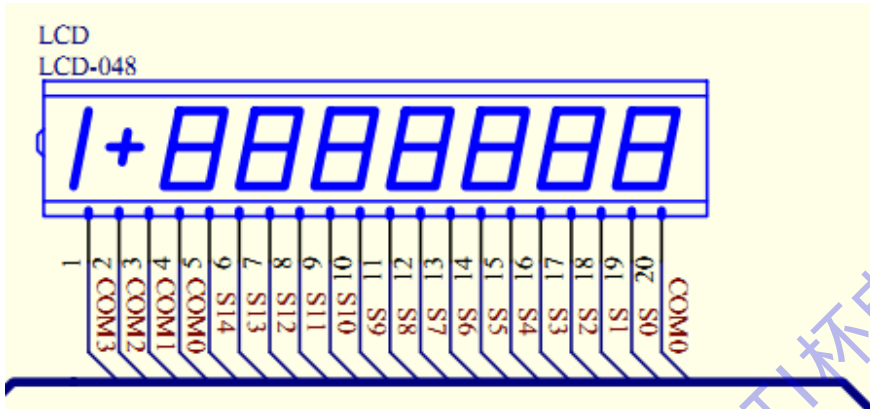


图 18

该模块由于实验箱内部集成有驱动，在输出方式设计上直接调用 `LCDMEM[i] = lcd_table[i];` 语句即可实现数据输出，需要注意的仅有刷新频率问题。

5.3.4. 模块流程

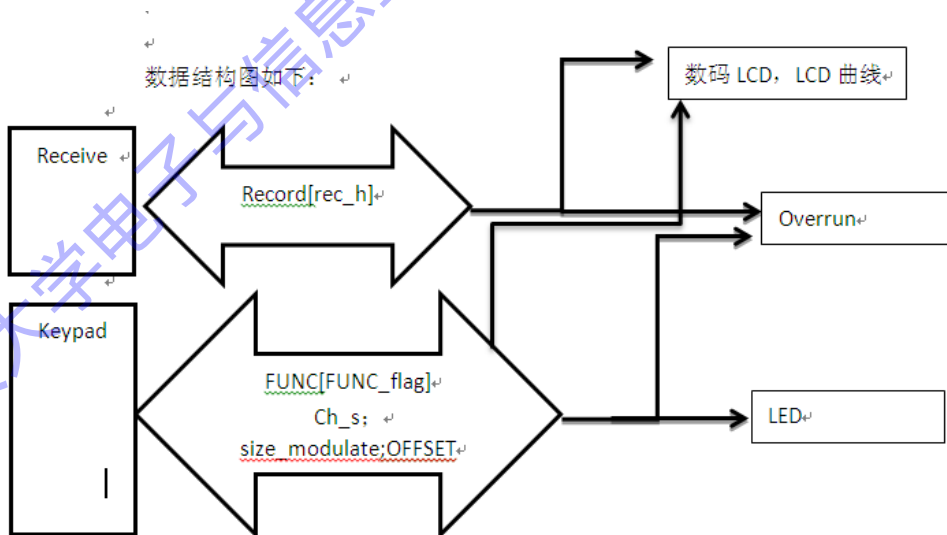


图 19

5.3.5 逻辑功能开发

逻辑功能开发，即利用逻辑运算以及不同程序结构（顺序，选择，循环）来实现特定功能。

- 接收模块

上位机接受模块是一切温度数据的产生源。

模块函数名: void recieve();

- 数制转换模块

该模块读取一个待显示的最新数据, 将其信息提取并存于几个变量中, 为下一级显示模块提供数据源。

模块函数名: void trans()

- 警报模块

该模块实现最新值与所设阈值的比较、判断, 并对应输出警报信号。

模块函数名: void alert ();

- 数值显示模块

模块将所要显示的数值在对应外设进行输出, 包括实时温度与阈值。

模块函数名: void print_rt()

- LCD 点阵显示模块

该模块实现 LCD 点阵的绘图功能。

模块函数名: LCD_fram(); LCD 固定框架描绘, 包括横纵坐标轴, 动态数值前的固定 LABEL

模块函数名: LCD_ch (); 刷新当前工作信道编号及信道新获取的数值

模块函数名: LCD_mod(); 类似 LCD_ch 输出当前放缩尺度与横轴温度, 详细分析略。

模块函数名: LCD_line();

- 键值处理模块

根据对应键值进行相应操作

模块函数名: switch (key_buff)

- 逻辑函数总表

函数名

recieve()

trans()

alert ()

输入数据	输出数据
BYTE TXRXBUFF[32]	int record[3][100] rec_h
record[ch_c][rec_h-1]	rec_o, rec_f, ten, one, tenth
rec_o, FUNC[2]	LED (8 或 0, led_Ctrl>>5) P6.7OUT=0 或 1

print_rt()	rec_o, rec_f, ten, one, tenth FUNC[flag]	LCDMEM[6~0]; LED (阈值, led_Ctrl>>0 或 1)
LCD_fram();		LCD 框架描绘
LCD_ch (); LCD_mod()	ch_s, ch_c, size_modulate, OFFSET, ten, one, tenth	display (输入值)
LCD_line();	record[ch_c][rec_h-1]	display_char(point[对应点], 对应行, 对应列)

表 2

5. 4. 温度控制模块

5. 4. 1. 设计目标

温控模块属于本系统中的附加功能模块，在原有的温度检测、数据发送及处理的基础上，实现对远端目标物体进行恒温控制，其温度控制的实时信息会在近段主控模块中通过数字和温度曲线显示，并且可以提供报警功能。

5. 4. 2. 系统框图

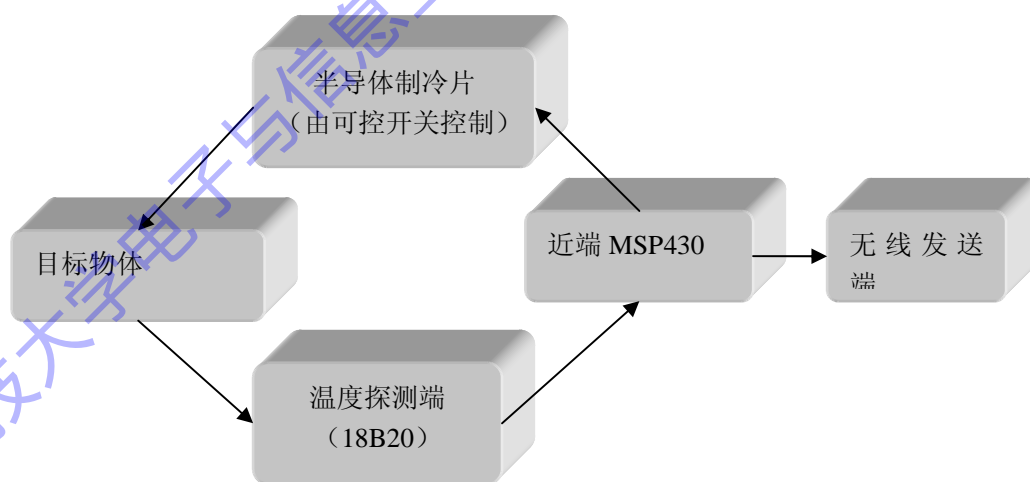


图 20

5. 4. 3. 温控模型

基于 TLE5206-2 的三段式 PD 温控算法

所谓三段是指从某一温度开始，向着目标温度逐渐逼近，直至稳定在目标温度的过程中所

经历的三个温控过程。首先，当温差高于 5 摄氏度时，我们可以全力进行加热或制冷向着目标温度靠近；温差低于 5 摄氏度高于 2 摄氏度，每个周期的占空比变为前一周期的 0.9 倍；当温差低于 2 摄氏度，则按照 PD 算法进行温控。下面将就三段中最核心的 PD 段中所用到的 PID 算法进行推导和分析。

PID (Proportional Integral Derivative) 控制是控制工程技术成熟、应用广泛的一种控制策略，经过长期的工程实践，已形成了一套完整的控制方法和典型的结构。

PID 工作机理：由于来自外界的各种扰动不断产生，要想达到现场控制对象值保持恒定的目的，控制作用就必须不断的进行。若扰动出现使得现场控制对象值(以下简称被控参数)发生变化，现场检测元件就会将这种变化采集后经变送器送至 PID 控制器的输入端，并与其给定值(以下简称 SP 值)进行比较得到偏差值(以下简称 e 值)，调节器按此偏差并以我们预先设定的整定参数控制规律发出控制信号，去改变调节器的开度，使调节器的开度增加或减少，从而使现场控制对象值发生改变，并趋向于给定值(SP 值)，以达到控制目的，如图 1 所示，其实 PID 的实质就是对偏差 (e 值) 进行比例、积分、微分运算，根据运算结果控制执行部件的过程。

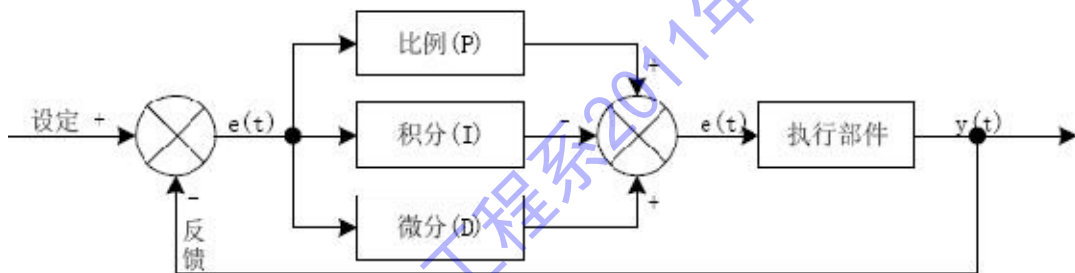


图 21 模拟 PID 控制系统原理图

PID 控制器的控制规律可以描述为：

$$\mu(t) = K_p \left[e(t) + \frac{1}{T_I} \int_0^t e(t) dt + T_D \frac{de(t)}{dt} \right] + \mu_0 \quad \text{式 (1)}$$

比例 (P) 控制能迅速反应误差，从而减小稳态误差。但是，比例控制不能消除稳态误差。比例放大系数的加大，会引起系统的不稳定。积分 (I) 控制的作用是：只要系统有误差存在，积分控制器就不断地积累，输出控制量，以消除误差。因而，只要有足够的时间，积分控制将能完全消除误差，使系统误差为零，从而消除稳态误差。积分作用太强会使系统超调加大，甚至使系统出现振荡。微分 (D) 控制可以减小超调量，克服振荡，使系统的稳定性提高，同时加快系统的动态响应速度，减小调整时间，从而改善系统的动态性能。根据不同的被控对象的控制特性，又可以分为 P、PI、PD、PID 等不同的控制模型。

数字 PID 的实现：由于计算机基于采样控制理论，计算方法也不能沿袭传统的模拟 PID 控制算法(如公式 1 所示)，所以必须将控制模型离散化，离散化的方法：以 T 为采样周期，k 为采样序号，用求和的形式代替积分，用增量的形式(求差)代替微分，这样可以将连续的 PID 计算公式离散：

$$t \approx kT \quad (k=0,1,2,\dots)$$

$$\int_0^t e(t) dt \approx T \sum_{j=0}^k e(jT) = T \sum_{j=0}^k e(j)$$

$$\frac{de(t)}{dt} \approx \frac{e(kT) - e[(k-1)T]}{T} = \frac{e_k - e_{k-1}}{T}$$

式 1 就可以离散为:

$$\mu_k = K_P [e_k + \frac{T}{T_I} \sum_{j=0}^k e_j + \frac{T_D}{T} (e_k - e_{k-1})] + \mu_0$$

或者:

$$\mu_k = K_P e_k + K_I \sum_{j=0}^k e_j + K_D (e_k - e_{k-1}) + \mu_0$$

这样就可以让计算机或者单片机通过采样的方式实现 PID 控制, 具体的 PID 控制又分为位置式 PID 控制和增量式 PID 控制, 公式 4 给出了控制量的全部大小, 所以称之为全量式或者位置式控制; 如果计算机只对相邻的两次作计算, 只考虑在前一次基础上, 计算机输出量的大小变化, 而不是全部输出信息的计算, 这种控制叫做增量式 PID 控制算法, 其实质就是求 $\Delta \mu$ 的大小, 而 $\Delta \mu_k = \mu_k - \mu_{k-1}$; 所以将式 4 做自减变换有:

$$\Delta \mu_k = \mu_k - \mu_{k-1}$$

$$= \mu_k = K_P [e_k - e_{k-1} + \frac{T}{T_I} e_k + \frac{T_D}{T} (e_k - 2e_{k-1} + e_{k-2})]$$

$$= K_P (1 + \frac{T}{T_I} + \frac{T_D}{T}) e_k - K_P (1 + \frac{2T_D}{T}) e_{k-1} + K_P \frac{T_D}{T} e_{k-2}$$

$$= A e_k + B e_{k-1} + C e_{k-2}$$

$$A = K_P (1 + \frac{T}{T_I} + \frac{T_D}{T}) \quad B = -K_P (1 + \frac{2T_D}{T}) \quad C = K_P \frac{T_D}{T}$$

在实际设计过程中, 我们发现在我们这个模型中, 真正需要避免的温度上升或下降中出现的过冲, 因此在这我们只需选用 PD 算法即可满足我们的需求。

另外我们控制制冷片加热和制冷的方式是由远端 MSP430 输出脉宽调制波形的高低电平, 控制 TLE5206-2 的输出电流 (正向、反向及零电流), 进而控制半导体制冷片的加热、制冷及停止。我们首先经由实验测定, 在不同目标温度下, 都有一个特定的占空比 (高电平加热低电平制冷) 能够保证目标物体在与外界相对热隔离 (由自制恒温箱实现) 的情况下能够使其相对稳定在这个温度下, 记此占空比为 α_0 , 对应 PID 算法中的 μ_0 。而后在实际温度偏离目标温度时我们需要设定一个校正占空比 $\Delta \alpha$, 从而控制温度向着目标温度趋近。由 PID 算法, $\Delta \alpha$ 可以由下式

确定

$$\Delta\alpha = -\beta[\Delta T_k + (\Delta T_k - \Delta T_{k-1})]$$

式中 ΔT_k 与 ΔT_{k-1} 分别为当前时刻和前一时刻的实际——目标温度差， β 为比例调节系数取正值，根据实际的硬件及物理条件测定。于是我们可以得到在 PD 算法的每一时刻占空比的表示式：

$$\alpha_k = \Delta\alpha + \alpha_0 = -\beta[\Delta T_k + (\Delta T_k - \Delta T_{k-1})] + \alpha_0$$

实际的程序实现中，我们只需在可调范围内每隔一摄氏度记下对应的 α_0 和 β ，形成一个数组，即可在设定好目标温度的情况下，由 MSP430 进行温度 PD 控制。

这里需要说明的是，以上 PD 算法由于需要制冷片电流能受控正负方向上导通，因此 TLE5206-2 能实现，并且在调试过程中，表现出较好的温控效果，温度抖动范围可以控制在 1 摄氏度以内。

由于器件问题，在本次硬件设计末期，我们紧急用 74HC245 及 5V 继电器对 TLE5206-2 进行了替代，由于继电器无法使半导体制冷片内电流可控双向流通，因此在低电平时，半导体制冷片对应的是停止工作状态，因此需要调整可调范围内每隔一摄氏度记下对应的 α_0 和 β ，最后在验收过程中，其表现出的温控指标不逊与 TLE5206-2，这也充分说明了三段式 PD 算法的适应性和稳定性。

5.4.4. 硬件设计

器件选型

在温控模块我们使用了三种器件分别为半导体制冷片 TEC12705，5V 电磁继电器以及 74HC245 驱动芯片

■ 半导体制冷片 TEC12705

经过多方面的查询资料，发现在弱电方面实现温度控制（既能制冷，又能加热）的器件只有半导体制冷器 TEC，半导体制冷片也叫热电制冷片，其原理是 Peltier 效应，它既可制冷又可加热，通过改变直流电流的极性来决定在同一制冷片上实现制冷或加热，这个效果的产生就是通过热电的原理来实现的。其实在原理上半导体制冷器只是一个热传递的工具。其优点为：

- 不需要任何制冷剂，可连续工作。
- 半导体制冷片具有两种功能，既能制冷，又能加热，制冷效率一般不高，但制热效率很高，永远大于 1。因此使用一个片件就可以代替分立的加热系统和制冷系统。
- 半导体制冷片是电流换能型片件，通过输入电流的控制，可实现高精度的温度控制，再加上温度检测和控制手段，便于组成自动控制系统。
- 半导体制冷片热惯性非常小，制冷制热时间很快，在热端散热良好冷端空载的情况下，通

电不到一分钟，制冷片就能达到最大温差。

- 半导体制冷片的温差范围，从正温 90℃到负温度 130℃都可以实现。。



图 22. TEC 制冷片

以下是该型号型号的 TEC 片的相关参数。

TEC12705	
描述	热电致冷器
最大直流电流	3.9A
最大直流电压	15.4V
最大产冷功率	33.4W
温度范围	-150℃~70℃
尺寸	40*40*4.7

■ H-桥电机驱动芯片 TLE5206-2

前面已经给出 TEC 片在工作的电压电流参数，所以我们知道应为了保证其正常工作，我们需要专门的供电器件来支持和控制 TEC 片的工作。在此我们选择了 TLE5206-2。

TLE5206-2 内部包括了功率 MOSFET 的 H 桥电路，具有较高的开关速度以及较高的噪声容限和抗干扰能力，驱动能力强，可直接驱动半导体制冷器工作。最大工作电压 40V，最大工作电流为连续电流 5A，峰值电流 6A；具有输出短路保护，过热保护；CMOS/TTL 电平输入；工作温度范围-40℃~150℃之间。

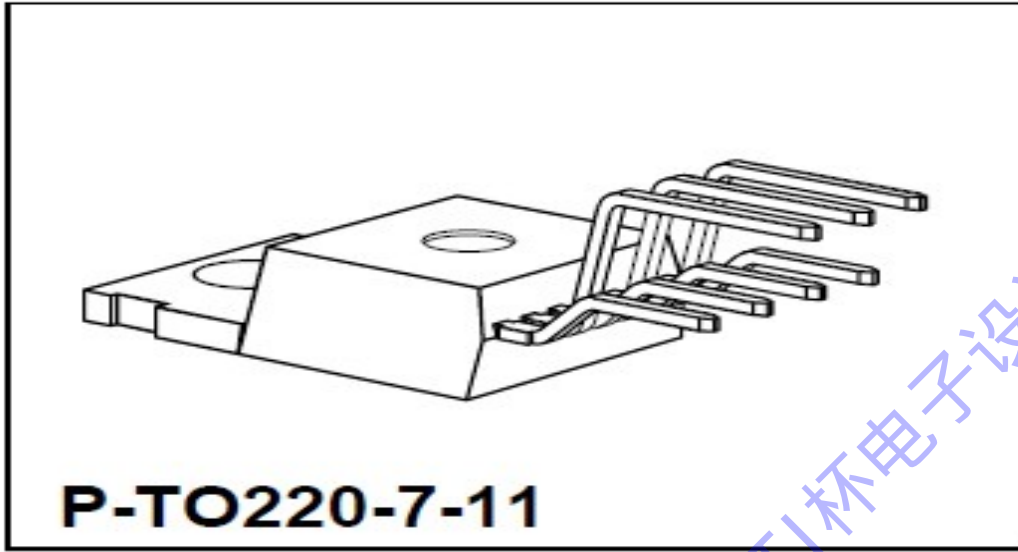


图 23. TLE5206-2

TLE 5206-2

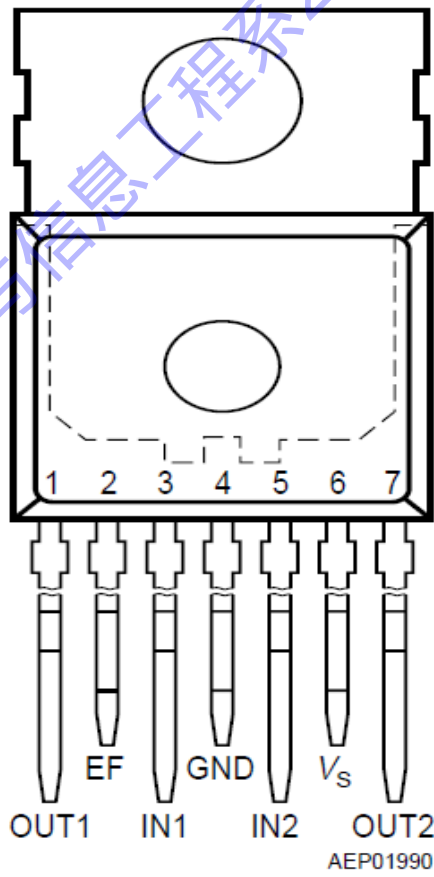


图 24

电路逻辑如下表:

IN1	IN2	OUT1	OUT2	Comments
L	L	L	L	Brake; both low side transistors turned-ON
L	H	L	H	Motor turns counterclockwise
H	L	H	L	Motor turns clockwise
H	H	H	H	Brake; both high side transistors turned-ON

表 3. 电路逻辑

■ 5V 电磁继电器以及 74HC245 驱动芯片

以上种器件都属于常见器件故不赘述，以下是两者的引脚图：

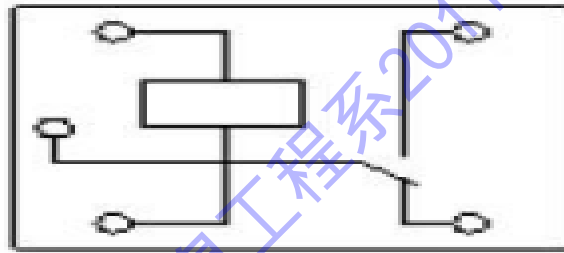


图 25. 5V 电磁继电器

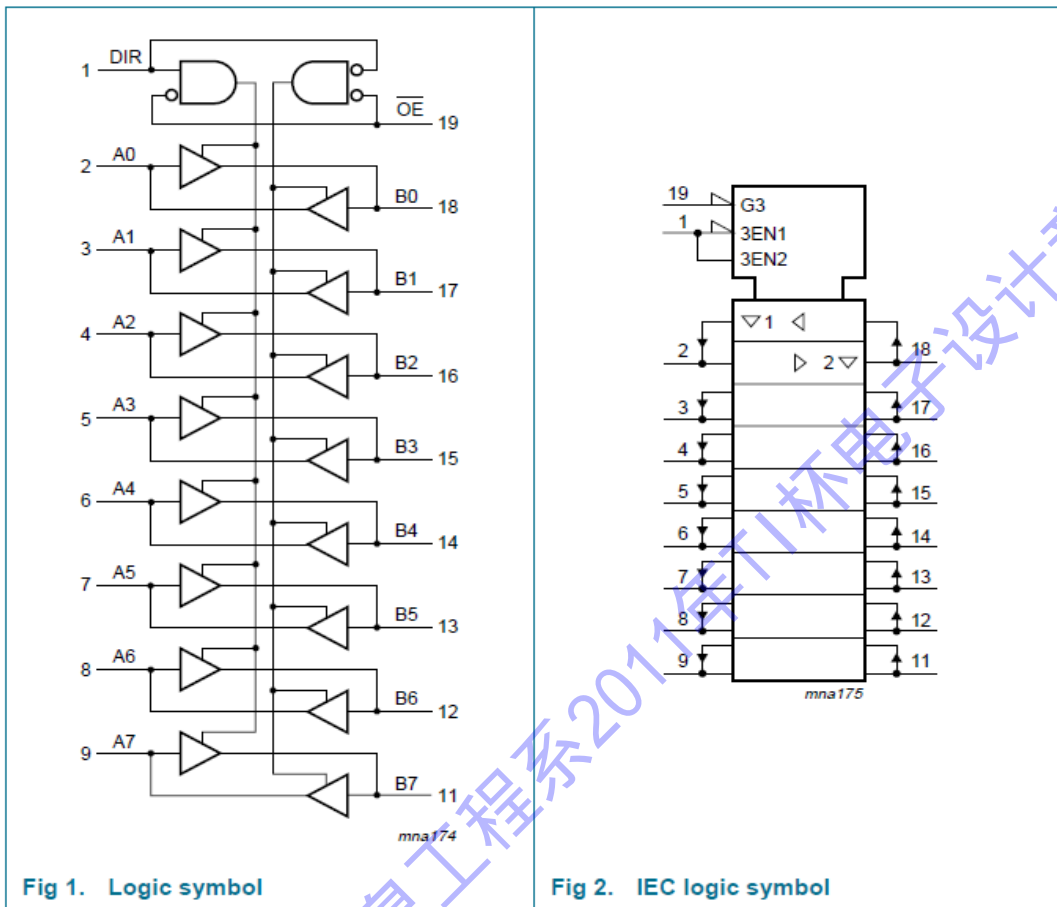


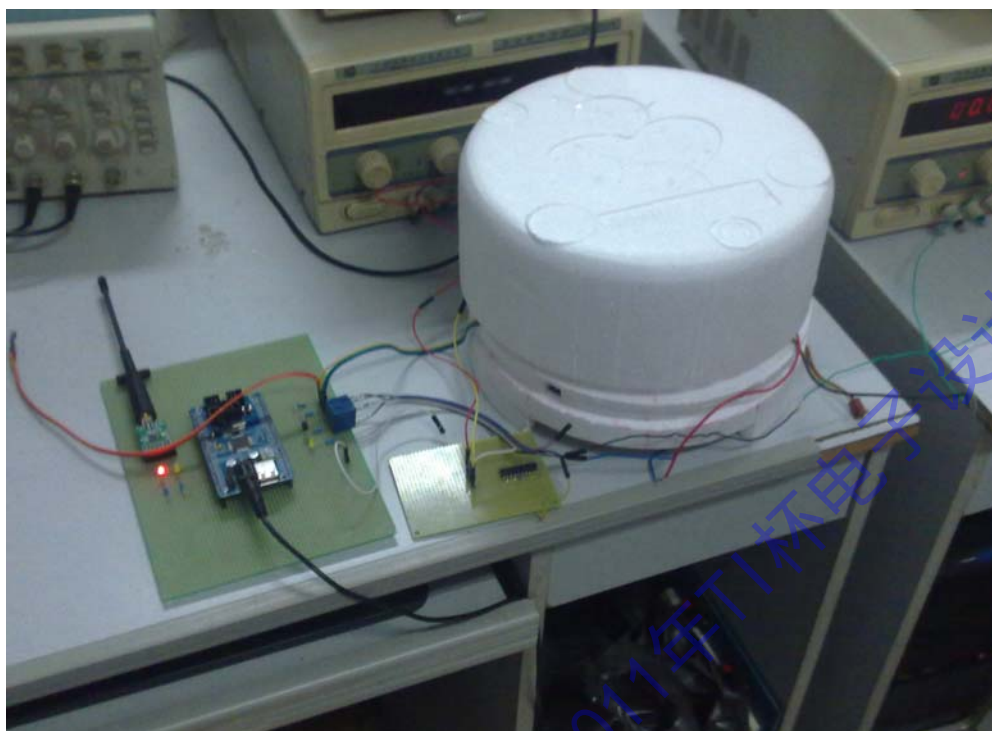
图 26. 74HC245

温控模块的搭建

根据各实验装置的特性和实验实际的需要，在与老师交流时，我们总结和吸收往届失败的经验教训，不仅在算法上有要求，而且在实验平台的搭建上，也要注意一下几个问题：

- 半导体制冷片电流转换不能过快，否则将导致制冷片寿命缩短，甚至损坏。
- 制冷片在工作一段时间后，热端的热量容易与冷端中和，一次在使用冷端时，应当及时将热端的热量导出。
- 在进行温控实验时，一定要在与外界温度相对隔离的条件下进行。
- 所选择的目标物体导热性能要好，否则温度测试的阻尼将会加大。
- 温度探测在物体上的位置也要适当选择，不能在过于偏离热源的地方，因为那样无法及时反映出物体的实际温度，同时也不能过于靠近热源，否则将会导致温度过冲现象严重。

根据以上所列注意事项，我利用身边的材料，之所了一个“热黑箱”，如下图所示：



华中科技大学电子与信息工程系2011年T1杯电子设计竞赛

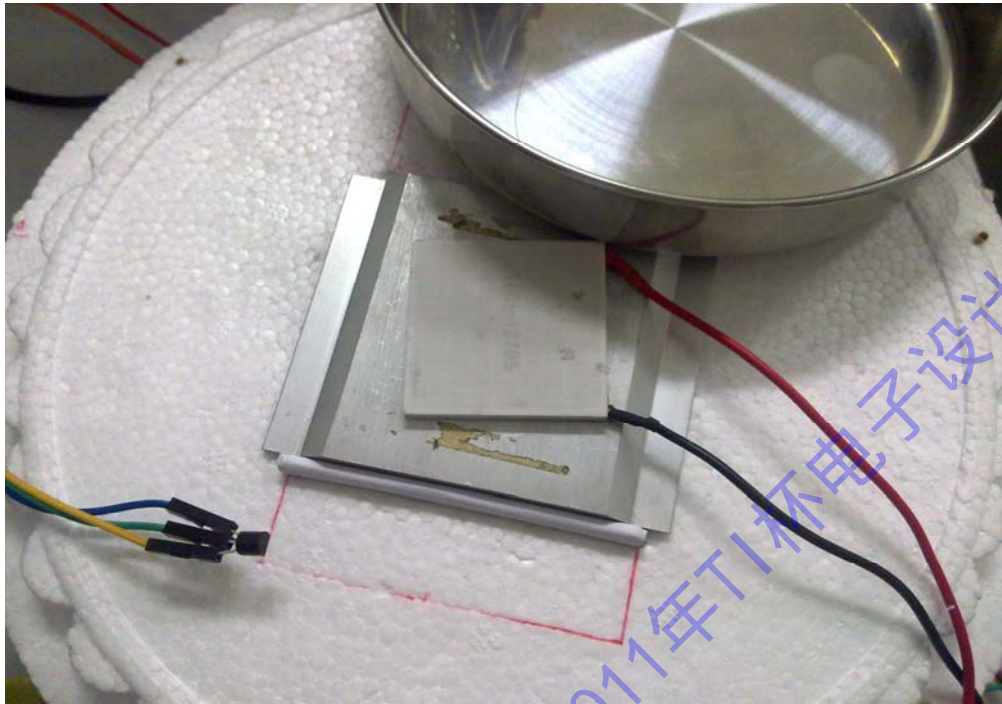


图 27 .平台搭建

如图所示，我们选择了一个不锈钢圆盘进行测量，半导体制冷片位于不锈钢圆盘下，在半导体制冷片下面是 CPU 专用的风扇散热片，为制冷片散热。温度探头的位置位于距圆心三分之二半径的地方。

5.4.5. 测试与调试

- 1) 确定个器件所允许通过的电压和电流值，选择适当的电路搭建方式、电源。
- 2) 确定 74HC245、5V 电磁继电器各引脚意义，并确定其都处于为损坏状态；探明半导体制冷片电流流向和每一面制冷或制热之间的关系，选择合适的工作面。
- 3) 搭建合适的测试电路，将各器件一次按照相应引脚相连，将 MSP430 的控制信号引入到 74HC245 上，通上小电源值，逐渐加大电源值，直至合适的电源值，中间确保各器件工作稳定。
- 4) 通过主控端的温度检测与显示装置，观察制冷片的工作状态，观察其是否按照设定的三段 PD 算法，随着温度的变化改变为适当的工作状态。
- 5) 结合实际的工作状态调整 β 和 α_0 值。
- 6) 在反复测试后，温控指标符合要求，进行制板工作。

6. 系统测试与结果

在各部分正确实现各自功能的基础上，我们进行了整机测试，具体步骤如下：
按照程序设计和模块特性，搭建整机系统，并下载代码程序。

● 无线模块测试:

- 1) 打开发送端（下位机）电源，观察到发送端提示灯不断闪烁，表明正在发送时数据发送。
- 2) 打开接收端（上位机）电源，观察到接收端提示灯不断闪烁，表明接收到数据，无线模块工作正常。

● 测温模块测试:

- 1) 观察接收机数字LCD的实时温度显示，观察到三路温度都在 27°C 的室温附近做细微变化，表明测温模块工作正常，接收端收到正确的温度数据。
- 2) 用手捏住一路的测温传感器，在接收端观察该路的温度变化情况，数字LCD屏上的温度数值逐渐升高，点阵LCD液晶显示屏绘制的曲线不断升高；松开手后温度数值和绘制的曲线都逐渐下降到室温，则该路测温模块正常工作。对二路，三路温度传感器做同样的操作，观察到上述现象，表明三路温度传感器都正常工作，而且上下位机和测温模块的兼容性正常。

● 主机功能模块测试:

该部分主要是调试按键功能的有效性。

- 1) 信道显示切换功能（按键1~3）：按下“1”号键，数字LCD上显示“CH1 27.5”（室温附近某一值），点阵LCD上显示“CH1 27.5”并绘制该路的温度曲线；按下“2”号键，数字LCD上显示“CH2 27.5”（室温附近某一值），点阵LCD上显示“CH2 27.5”并绘制该路的温度曲线；按下“3”号键，数字LCD上显示“CH2 27.5”（室温附近某一值），点阵LCD上显示“CH2 27.5”并绘制该路的温度曲线，表明三个按键的功能正常，能够实现信道的切换。
- 2) 巡检功能（‘#’键）：按下‘#’号键，开始巡检模式。观察到数字LCD和点阵LCD显示的信道每个大约1.3秒切换一次，三路循环显示，表明巡检模式工作正常。
- 3) 点阵LCD的动态显示功能（按键0, 4~6）：本系统设计的点阵LCD具有精度和纵轴动态可调的功能。对1路（其他路也可）温度传感器加温，按下‘4’号键，可以发现曲线幅度明显降低，升温变得十分平缓，这是由于精度太低而升温速度很慢造成的。按下‘5’号键，可以发现曲线幅度按指数背的增加，升温变化曲线十分明显。这是由于纵轴的精度提高，显示效果增强，以上现象表明曲线显示的精度变换功能有效。按下‘6’号键，可以观察到LCD曲线整体下移，显示屏上的中轴值逐渐减‘1’；按下‘0’号键，LCD曲线整体上移，显示屏上的中轴值逐渐加‘1’，表明纵轴的调节功能正常。
- 4) 阈值调节功能（按键7~9）：阈值包括上阈值和下阈值。阈值的调节由7~9键来完成。按下‘7’号键，6位七段码LED上显示对以信道（具体是那信道有1~3信道选功能键控制）的上阈值（下阈值），按‘8’号键，上阈值（下阈值）逐渐增加，增加步长为 1°C ；按下‘9’号键，上阈值（下阈值）的值逐渐减小，步长也是 1°C 。再按下‘7’号键，进入下阈值（上阈值）的调节模式。我们观察到当增加下阈值的值达到上阈值的使在增加数值也不会增加，而在减小上阈值时，当下降到下阈值的值时也不能再向下减小。这正是我们设置的保护功能，上阈值不能低于下阈值。上述现象表明阈值调节功能正常。

- 5) 超限报警功能：设定信道‘1’的上阈值为 30°C ，用手对‘1’路探头加温，发现当加热的温度超过 30°C 使，LED 七段数码管的第二位由全灭变为全亮同时蜂鸣器发出报警声，表明超限声光报警正常。

● 温控模块

温控模块的指标完全达到了预先目标，即上下抖动范围在 5 摄氏度以内，实际上我们的指标达到了 1 摄氏度，远优于基本要求。在逼近目标温度的过程中，没有出现过严重过冲现象，并且能很快完成逼近并很快实现温度稳定。在 10 摄氏度至 40 摄氏度的目标温度范围内，在距离目标温度 6 摄氏度的情况下基本都可以在 3 分钟完成达到并稳定在目标温度。总体而言，温控模块较为出色地完成了其功能。

● 远距离测试：

发送端保持不动，移动接收端的位置，增大发送端与接收端的距离，将发送端和接收端放置在实验室的对角处，两端同时工作时，在接收端任然能够接收到正确的数据。

● 系统稳定性测试

- 1) 静态稳定性测试：重新启动发送和接收端，保持两端的状态不变，让其长时间正常工作。我们等待时间为 20 分钟后观察系统的工作情况，发现系统的按键，绘制曲线，升温曲线等所有功能都正常，表明在较长的时间范围内，系统能够正常工作，稳定性较高。
- 2) 动态稳定性测试：在 12 步的基础上，给温度传感器升温，在接收端反复切换显示状态并不断随机设置阈值、精度等，发现频繁切换功能不影响系统的稳定性。

7. 结束语

综上所述，我们的系统不仅完满地实现了各种基本功能，而且还实现了难度较大的温度控制和温度曲线显示功能。在各种测试条件下保持了性能的稳定。因而，这是一套相对优秀的系统。

对于我们小组而言这次硬件设计的过程是十分曲折的，中间的所走过的弯路，所经历的挫败、失望、彷徨直到坚持中突然的“起死回生”，在过去两个星期的每一天，融入在我们每一行代码中，每一根导线里，我们无心也无法去过多思考这零零总总的琐碎细节。然而，今天当我们回望这一切，欣慰夹杂着感慨便涌上心头。

从购买器件过程中商家反复出售问题芯片，到焊好的电路板却无法工作，再到高级功能温控模块在最后一天因为芯片质量问题而不得不重新修订方案等等，这些打击现在看来却是我们硬设路上一个个里程碑，让我们倍感鼓舞。如果我们没有当时大胆的规划与分工，不必这些打击来掺和，我们或许会在自己的手忙脚乱中葬送这次硬设；如果我们没有那份坚持与执着，或许我们早就在一次次困境面前彷徨，犹豫最终放弃温控模块；如果我们没有那些在困境时的相互鼓舞与通力合作，很难想象我们是怎么在最后一天里面面对片子的突然失灵，从零开始，重新设计并实现了一个模块。

一次硬件设计留给我们的除了知识与技能之外，更为重要的是一份态度，一种品质。

8. 参考文献

- 【1】 兰建军,隋立东,崔玉梅. DS18B20 在多点测温系统中的应用. 东北电力学院学报, 2003-12
- 【2】 王超, 魏军, 王玉. 多路温度巡检仪. 南京师范大学电气与自动化工程学院
- 【3】 张长利, 李永常. 多路温度单片机监控系统的研究. 东北农学院学报, 1992-03(01)
- 【4】 宋丹. 多路温度巡检系统的设计与应用. 仪器仪表装置. 天津 300192
- 【5】 周四清, 罗雪峰, 何献华, 基于 DS18B20 的八路温度巡检仪的设计与应用. 科技创新导报, 2008(13)
- 【6】 雷民. 基于 DS18B20 的温度巡检系统的设计与实现. 华中科技大学硕士学位论文. 光电工程, 2006 年
- 【7】 尹溪, 万曼华, 王晓军. 基于单片机的温度巡检仪系统. 电工文摘. 2008-05
- 【8】 刘毅力, 郭昆丽, 邵文权. 多路温度巡检仪与 PC 机的 RS-485 通讯. 西安工程大学电子信息学院. 西安工程科技学院学报, 2007-04
- 【9】 张琦, 张英. PC 机与 MSP430 单片机串行通信的实现方法. 解放军理工大学工程兵工程学院. 中国工程机械学报, 2007-01(05)
- 【10】 秦龙. MSP430 单片机 C 语言应用程序设计实例精讲(光盘)一版. 电子工业出版社. 2006-05

9.

10.

11.

12.

6.

7.

8.

9.

10.

11.

丁
丁
丁

华中科技大学电子与信息工程系2011年TI杯电子设计竞赛