

《物联网技术及应用实验》

实验报告本

|  |  |
| --- | --- |
| 班级： | **信工211** |
| 学号： | **21012909** |
| 姓名： | **孟依然** |
| 指导教师： | **黄如** |

信息科学与工程学院

2024年04月

**实验四** 温湿度传感器采集实验

1. **实验目的**

（1）掌握温湿度传感器的操作方法；

（2）掌握温湿度传感器采集程序的编程方法。

**二、实验装置**

硬件：计算机一台（操作系统为Windows XP或Windows 7）；CVT-IOT-VSL实验箱一台；CC DEBUGGER仿真器；USB数据线一根。

软件：IAR Embedded Workbench for MCS-51开发环境。

**三、基础知识**

（1）温度传感器介绍

采用SHT10温度传感器，SHTxx 系列单芯片传感器是一款含有已校准数字信号输出的温湿度复合传感器。它应用专利的工业COMS过程微加工技术（CMOSens®），确保产品具有极高的可靠性与卓越的长期稳定性。传感器包括一个电容式聚合体测湿元件和一个能隙式测温元件，并与一个14 位的A/D 转换器以及串行接口电路在同一芯片上实现无缝连接。因此，该产品具有品质卓越、超快响应、抗干扰能力强、性价比极高等优点。

每个 SHTxx 传感器都在极为精确的湿度校验室中进行校准。校准系数以程序的形式储存在OTP 内存中，传感器内部在检测信号的处理过程中要调用这些校准系数。

两线制串行接口和内部基准电压，使系统集成变得简易快捷。超小的体积、极低的功耗，使其成为各类应用甚至最为苛刻的应用场合的最佳选则。产品提供表面贴片 LCC（无铅芯片）或4 针单排引脚封装。特殊封装形式可根据用户需求而提供。

（2）温湿度传感器的接口电路

温湿度传感器的接口电路如图4-1所示，引脚分配如表4-1所示。通过CC2530的I/O口仿真 SHT10要求的串行接口时序，以读出SHT10温湿度传感器采集的当前的温度和湿度值。

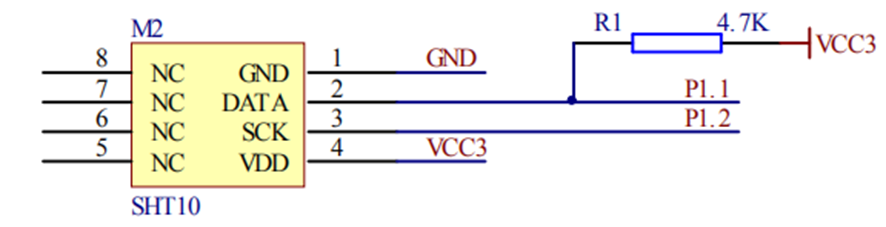


图4-1 温湿度传感器接口电路

表4-1 SHT10引脚分配

|  |  |  |
| --- | --- | --- |
| 引脚 | 名称 | 描述 |
| 1 | GND | 地 |
| 2 | DATA | 串行数据，双向 |
| 3 | SCK | 串行时钟，输入口 |
| 4 | VDD | 电源 |
| 5-8 | NC | 必须为空 |

（3）实验说明

本实验通过以设定的时间间隔，循环采集温湿度传感器的温度和湿度，并通过串口在串口调试助手中显示出来。

①函数 void Sensor\_PIN\_INT(void)：读写温湿度传感器实验时，对相应的 I/O 口进行配置。

②函数 uint32 ReadSHT1(uint8 Addr)；为读取温湿度传感器值的调用函数。

函数功能:返回 SHT10 温湿度芯片采集的值

输入参数:地址值（根据芯片手册，参数为温度地址 0x3 或湿度地址 0x5）

**四、实验内容**

（1）实验箱上电，使用仿真器连接好带温湿度模块的CC2530节点；

（2）参照 IAR 安装及使用说明中的步骤“如何新建一个工程->添加配置->添加文件->编译链接->下载调试运行”的过程，新建一个工程SHT10，添加相应的文件，并修改SHT10的工程设置；

（3）创建SHT10.c 并加入到工程SHT10中；

（4）编写SHT10相关函数，在设置的间隔时间循环显示温度和湿度的值，并通过串口发送出来；

#define uint8 unsigned char

#define uint16 unsigned int

#define uint32 unsigned long

#include "ioCC2530.h"

#define SHT1DATA\_HIGH P1 |= 0x02

#define SHT1DATA\_LOW P1 &= 0xFD

#define SHT1SCK\_HIGH P1 |= 0x04

#define SHT1SCK\_LOW P1 &= 0xFB

#define SHT1READY ((P1>>1)&0x1)

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

函数声明

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

void Sensor\_PIN\_INT(void);

uint16 ReadAdcValue(uint8 ChannelNum,uint8 DecimationRate,uint8 RefVoltage);

void SHT1\_Reset(void);

void SHT1\_Start(void);

void SHT1\_SendAck(void);

void SHT1\_WriteCommandData(uint8);

uint8 SHT1\_ReadData(void);

uint8 SHT1\_Ready(void);

void SHT1\_WriteReg(uint8);

uint16 SHT1\_ReadReg(void);

void SHT1\_INT(void);

uint32 ReadSHT1(uint8 Addr);

uint8 ReadTc77(void);

void SET\_ADC\_IO\_SLEEP\_MODE(void);

void SET\_ADC\_IO\_ADC\_MODE(void);

extern void UartTX\_Send\_String(uint8 \*Data,int len);

uint8 CRC8(uint8 crc, uint8 data);

void Delay(void);

void Delay10ms(uint8 Times);

/\*函数功能:读出AD口的数据

输入参数:ChannelNum:采集的通道号 0-0xF

1000: AIN0–AIN1

1001: AIN2–AIN3

1010: AIN4–AIN5

1011: AIN6–AIN7

1100: GND

1101: Reserved

1110: Temperature sensor

1111: VDD/3

DecimationRate:分辩率 00: 64 decimation rate (7 bits ENOB)

01: 128 decimation rate (9 bits ENOB)

10: 256 decimation rate (10 bits ENOB)

11: 512 decimation rate (12 bits ENOB)

RefVoltage:参考电压:00: Internal reference

01: External reference on AIN7 pin

10: AVDD5 pin

11: External reference on AIN6–AIN7 differential input

返回值:16bit的采集数据

\*/

uint16 ReadAdcValue(uint8 ChannelNum,uint8 DecimationRate,uint8 RefVoltage)

{

uint16 AdValue;

if(ChannelNum == 0xe){//片内温度到ADC\_SOC

TR0 = 1;

ATEST = 1;

}

else{

TR0 = 0;

ATEST = 0;

}

ADCCON3 = ChannelNum&0xf;

ADCCON3 = ADCCON3 | ((DecimationRate&0x3)<<4);

ADCCON3 = ADCCON3 | ((RefVoltage&0x3)<<6);

ADCCON1 = ADCCON1 | (0x3<<4);//ADCCON1.ST = 1时启动

AdValue = ADCL; //清除EOC

AdValue = ADCH;

ADCCON1 = ADCCON1 | (0x1<<6);//启动转换

while(!(ADCCON1&0x80));

AdValue = ADCH;

AdValue = (AdValue<<6) + (ADCL>>2);

ADCCON1 = ADCCON1 & 0x7f;

return AdValue;

}

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

void Sensor\_PIN\_INT(void)

传感器及ADC I/O口初始化.

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

void Sensor\_PIN\_INT(void)

{

//用于温湿度测量

P1INP &= (~(0x1 | (0x1<<1) | (0x1<<2) | (0x1<<6) | (0x1<<7)));//P1.0,P1.1,P1.2,P1.6,P1.7上拉

P1SEL &= ~((1<<1)|(1<<2));//P1.1,P1.2为GPIO

P1DIR |= (1<<1)|(1<<2);//P1.1,P1.2为OUTPUT

}

//当uC和SHT10通信中断时,复位通信口

void SHT1\_Reset(void)

{

uint8 i;

SHT1DATA\_HIGH;

for(i=0;i<11;i++){

Delay();

SHT1SCK\_LOW;

Delay();

SHT1SCK\_HIGH;

Delay();

}

}

//传输启始信号

void SHT1\_Start(void)

{

SHT1DATA\_HIGH;

SHT1SCK\_LOW;

Delay();

SHT1SCK\_HIGH;

Delay();

SHT1DATA\_LOW;

Delay();

SHT1SCK\_LOW;

Delay();

SHT1SCK\_HIGH;

Delay();

SHT1DATA\_HIGH;

Delay();

SHT1SCK\_LOW;

}

//为0时,写命令正确;为1时错误

uint8 SHT1\_Ready(void)

{

//读应答信号

P1DIR &= ~(1<<1); //P11为INPUT

Delay();

return(SHT1READY);

}

void SHT1\_SendAck(void)

{

SHT1SCK\_HIGH;

Delay();

SHT1SCK\_LOW;

Delay();

}

//为0时,写命令正确;为1时错误

void SHT1\_WriteCommandData(uint8 bCommand)

{

uint8 i;

Delay();

SHT1SCK\_LOW;

Delay();

for(i=0;i<8;i++){

if(bCommand&(0x1<<(7-i)))

SHT1DATA\_HIGH;

else

SHT1DATA\_LOW;

Delay();

SHT1SCK\_HIGH;

Delay();

SHT1SCK\_LOW;

}

P1DIR &= ~(1<<1); //P11为INPUT

}

//读一个字节的数据

uint8 SHT1\_ReadData(void)

{

uint8 i,bResult;

bResult = 0;

P1DIR &= ~(1<<1); //P11为INPUT

Delay();

for(i=0;i<8;i++){

if(SHT1\_Ready() != 0)

bResult |= (0x1<<(7-i));

Delay();

SHT1SCK\_HIGH;

Delay();

SHT1SCK\_LOW;

}

P1DIR |= (1<<1); //P11为OUTPUT

return bResult;

}

//写状态寄存器

void SHT1\_WriteReg(uint8 Value)

{

while(1){

SHT1\_Start();

SHT1\_WriteCommandData(6);

if(SHT1\_Ready() != 0){ //无应答

SHT1\_Reset();

continue;

}

else{

P1DIR |= (1<<1); //P11为OUTPUT

SHT1DATA\_LOW;

SHT1\_SendAck();

}

SHT1\_WriteCommandData(Value);

if(SHT1\_Ready() != 0){ //无应答

SHT1\_Reset();

continue;

}

else{

SHT1DATA\_LOW;

SHT1\_SendAck();

}

break;

}

}

uint16 SHT1\_ReadReg(void)

{

uint16 lResult;

while(1){

SHT1\_Start();

SHT1\_WriteCommandData(7);

if(SHT1\_Ready() != 0){ //无应答

SHT1\_Reset();

continue;

}

else{

SHT1DATA\_LOW;

SHT1\_SendAck();

break;

}

}

lResult = (SHT1\_ReadData()<<8);

SHT1DATA\_LOW;

SHT1\_SendAck();

lResult |= SHT1\_ReadData();

SHT1DATA\_HIGH;

SHT1\_SendAck();

return lResult;

}

uint32 ReadSHT1(uint8 Addr)

{

uint32 lResult;

while(1){

SHT1\_Start();

SHT1\_WriteCommandData(Addr);

if(SHT1\_Ready() != 0){ //无应答

SHT1\_Reset();

continue;

}

else{

SHT1\_SendAck();

break;

}

}

Delay10ms(60);

lResult = SHT1\_ReadData();

lResult = lResult<<16;

SHT1DATA\_LOW;

Delay();

SHT1\_SendAck();

lResult |= ((uint16)SHT1\_ReadData()<<8);

SHT1DATA\_LOW;

Delay();

SHT1\_SendAck();

lResult |= SHT1\_ReadData();

SHT1DATA\_HIGH;

Delay();

SHT1\_SendAck();

return lResult;

}

uint8 CRC8(uint8 crc, uint8 data)

{

uint8 i;

crc = crc ^data;

for (i = 0; i < 8; i++)

{

if ((crc & 0x01) != 0) crc = (crc >> 1) ^ 0x8c;

else crc = crc >> 1;

}

return crc;

}

void Delay(void)

{

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

}

void Delay10ms(uint8 Times){

uint8 i;

uint16 j;

for(i=0;i<Times;i++){

for(j=0;j<5000;j++){

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

}

}

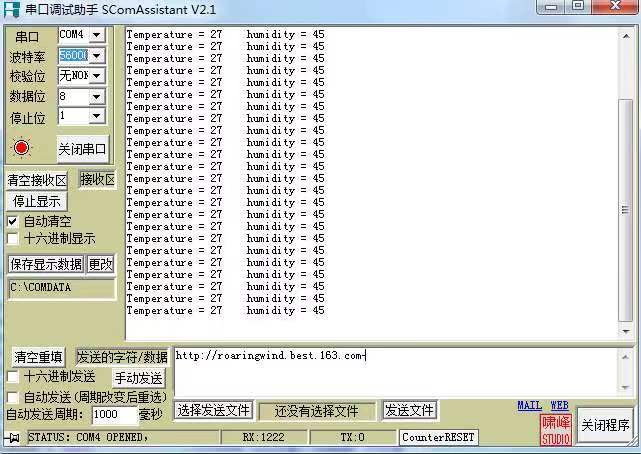
}

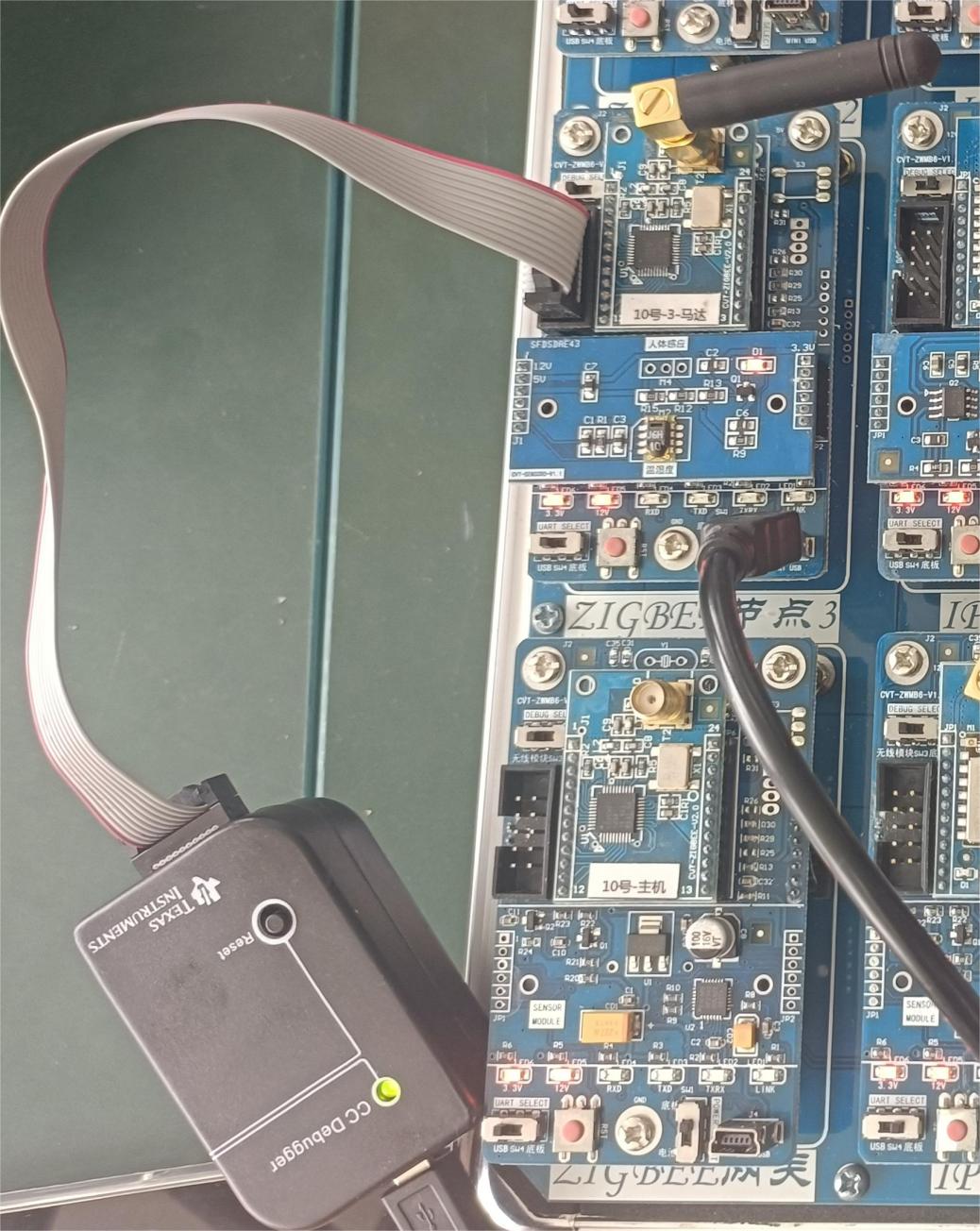
（5）编译SHT10，成功后，下载并运行，通过串口调试助手观察温湿度值。

四**、实验结果**

将温湿度模块插好，实验箱通电后，将纸板放在温湿度模块上，显示屏上的结果如图所示

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 节点名称 | 节点类型 | 源地址 | 父地址 | 温度 | 湿度 |
| 第一次 | 温湿度传感器 | 终端节点 | OXE74A | OX8723 | 27 | 45 |
| 第二次 | 温湿度传感器 | 终端节点 | OXE74A | OX8723 | 27 | 45 |
| 第三次 | 温湿度传感器 | 终端节点 | OXE74A | OX8723 | 27 | 45 |
| 第四次 | 温湿度传感器 | 终端节点 | OXE74A | OX8723 | 27 | 45 |
| 第五次 | 温湿度传感器 | 终端节点 | OXE74A | OX8723 | 27 | 45 |





**实验四成绩：\_\_\_\_\_\_\_\_\_\_\_\_ 教师签名：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**