



单总线多点高精度测温解决方案

概述：

在传统模拟信号远距离温度测量系统中，需要很好地解决引线误差补偿、多点切换误差和放大电路零点漂移误差等技术问题，才能够达到较高的测量精度。另外一般监控现场的电磁环境都非常恶劣，各种干扰信号较强，模拟温度信号容易受到干扰而产生测量误差，影响测量精度。因此，在温度测量系统中，采用抗干扰能力强的新型数字温度传感器是解决这些问题的最有效方案。相对于传统模拟测温系统，杭州晶华微电子有限公司研发的新型数字温度传感器 SD5820 具有体积更小、精度更高、适用电压更宽、采用单总线、可组网等优点，在实际应用中取得良好的测试效果。应用领域包括温控系统、工业过程控制、电源热保护、环境温度检测等。

SD5820 的主要特点：

- 1、兼容 DS18B20 芯片。
- 2、工作电压范围：2.7V~5.5V，
- 3、温度精度： $-10^{\circ}\text{C} \sim +85^{\circ}\text{C}$ 范围典型误差 $\pm 0.5^{\circ}\text{C}$ ， $-55^{\circ}\text{C} \sim +125^{\circ}\text{C}$ 范围典型误差 $\pm 0.8^{\circ}\text{C}$ 。
- 4、温度分辨率为 12 位，对应的分辨温度为 0.0625°C ，可实现高精度测温，完成一次温度测量的时间约为 750ms。
- 5、测量结果直接输出数字温度信号给 CPU，带 CRC 校验功能，具有极强的抗干扰和纠错能力。
- 6、简单的单总线接口方式，SD5820 在与微处理器连接时仅需要一条通讯线即可实现双向通信。也可以通过 DIO 信号产生寄生电源方式给芯片供电。
- 7、支持多点组网功能，多个 SD5820 可以并联在唯一的 DIO 线上，实现组网多点测温。
- 8、SD5820 芯片把全部传感器及转换电路集成在形如一只三极管的集成电路内；如果是多个芯片并联使用 DIO 寄生电源单线供电时，需使用外部强上拉 PMOS 器件辅助通讯可实现高精度测温。

SD5820 的应用电路：

SD5820 测温系统具有测温系统简单、测温精度高、连接方便、占用口线少等优点。下面是 SD5820 几个不同应用方式下的电路图：

如图 1、图 2 所示，在外部电源供电方式下，SD5820 工作电源由 VDD 引脚接入，每个芯片 VDD 引脚加一个 1uF 滤波电容就可以保证转换精度。可以同时在线路上挂接多个 SD5820 传感器，组成多从机测温系统。

在外部电源供电方式下，上拉电阻可选择 1k~10k，电容可选择 1uF~10uF，实际推荐上拉电阻 4.7k，电容 1uF。

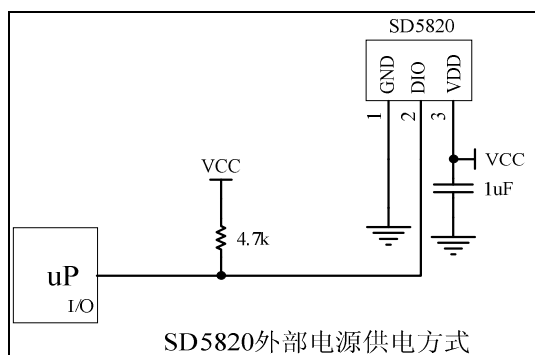


图 1. SD5820 外部电源供电方式图

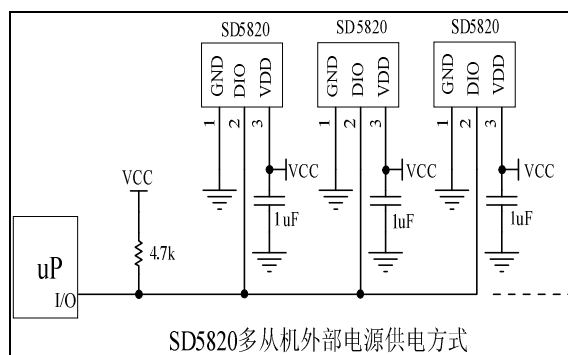


图 2. SD5820 多从机外部电源供电方式图

在寄生电源供电方式下，SD5820 从单线信号线上汲取能量：在信号线 DIO 处于高电平期间能量储存在内部电容里，在信号线处于低电平期间消耗内部电容上的电能工作，直到高电平到来再给寄生电源电容充电。要想使 SD5820 进行精确的温度转换，DIO 线必须保证在温度转换期间提供足够的能量。

使用寄生电源方式供电有三个好处：

- 1、进行远距离测温时，无需本地电源；
- 2、可以在没有常规电源的条件下读取数据；
- 3、电路更加简洁，仅使用一根 I/O 口实现测温。

图 3 为单个芯片采用寄生电源供电方式电路图（PMOSFET 可以不用）。

SD5820 多从机寄生电源供电方式如图 4 所示。由于每个 SD5820 在温度转换期间工作电流为 170uA 左右，当多个温度传感器挂在同一根 I/O 线上进行多点测温时，只靠 4.7k 上拉电阻无法迅速补充能量，导致测温异常。为此，MCU 在输出高电平或者检测到输入高电平时，可通过 MCU 的另一个端口控制导通 PMOSFET，把 DIO 线强上拉到 VCC。由于 PMOSFET 的导通阻抗很低，可以迅速给每一个芯片的 VDD 补充电能，维持高精度测温。当 MCU 输出低电平或者检测到输入低电平时，应先断开 PMOSFET。

在寄生电源供电方式下，使用 VCC 电源 3.3V ~ 5.5V，仍然保证在 -55℃ ~ +125℃ 范围内的测温精度。其中，PMOSFET 选择导通阻抗小，开关快的管子，建议在其栅极接一个 100k 的上拉电阻至 VCC，使 PMOSFET 不易损坏，（加 100k 电阻防止 PMOSFET 栅极悬空，给栅极和源极放电，默认状态 PMOSFET 处于断开状态）。上拉电阻可选择 1k~10k，推荐使用上拉电阻 4.7k。

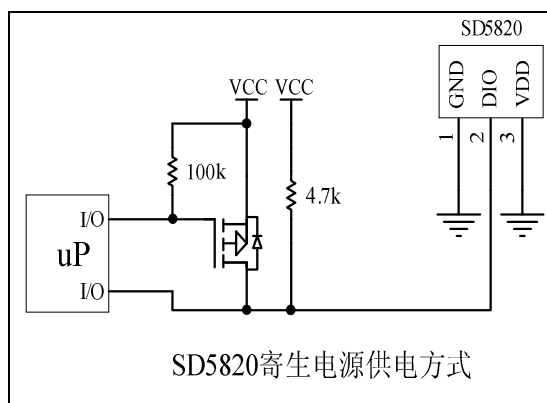


图 3. SD5820 寄生电源供电方式图

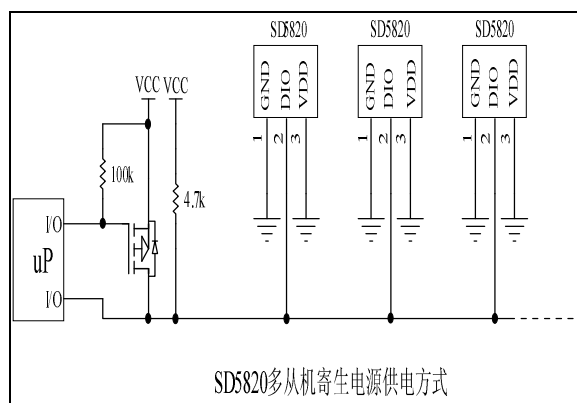


图 4. SD5820 多从机寄生电源供电方式图

设计注意事项:

1、较小的硬件开销需要相对复杂的软件进行补偿，由于 SD5820 与微处理器间采用串行数据传送，因此，在对 SD5820 进行读写编程时，必须严格的保证读写时序，否则将无法读取测温结果。

2、SD5820 初次上电需最大 6S 的启动时间（寄生供电，DIO 上拉电阻 4.7k 到 VDD=3.0V），电压升高或者减小上拉电阻能够缩短启动时间。只有初次上电需要启动时间，后续上电过程就不需要上电启动时间。

3、在由外部电源供电的多从机系统（图 2）中：

SD5820 数量越多，DIO 节点的寄生电容和电阻越大，DIO 信号恢复至高电平需要时间越长，可能导致通讯异常。所以上拉电阻值不能太大。

当 SD5820 数量较多时，建议使用外部强上拉 PMOS 器件辅助通讯（如图 4 的 PMOSFET）。

从机 SD5820 上传数据时的低电平 Sink 能力为 5mA (typ.)，所以上拉电阻不能小于 1K。

4、在由寄生电源供电的多从机系统（图 4）中：

必须配合使用外部强上拉 PMOSFET，以迅速补充 VDD 上的电能。

当 SD5820 数量 N 值较大时，经过强上拉 PMOSFET 的平均充电电流=0.2mA*N 也很大，瞬态充电电流更大（对电容 $C_{pp}=10\mu F*N$ 充电）。这要求 PMOSFET 的导通阻抗(Ron)尽量小，电源 VCC 的负载能力足够强，避免干扰到 MCU。

经实验证明，将 32 个多从机并联 DIO 并使用寄生电源供电，在 $-55^{\circ}\text{C} \sim +125^{\circ}\text{C}$ 范围内测温 1 小时都能正常，数据参考下图。测试时电源电压 3.3V，上拉电阻 4.7k。最多并联使用的数量，由用户电路系统的实际环境和参数决定。

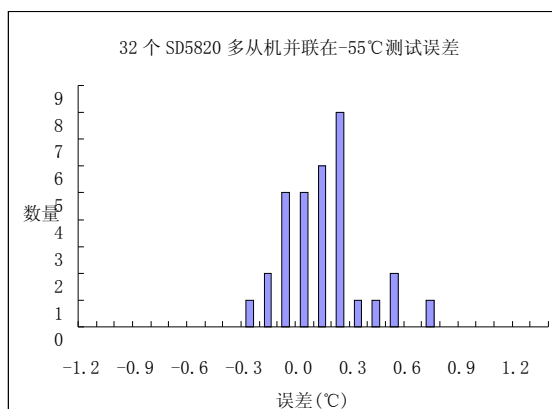


图 5. 32 个 SD5820 多从机并联在 -55°C 测试误差

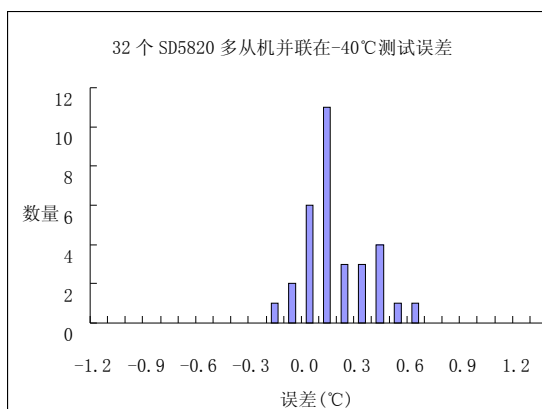


图 6. 32 个 SD5820 多从机并联在 -40°C 测试误差

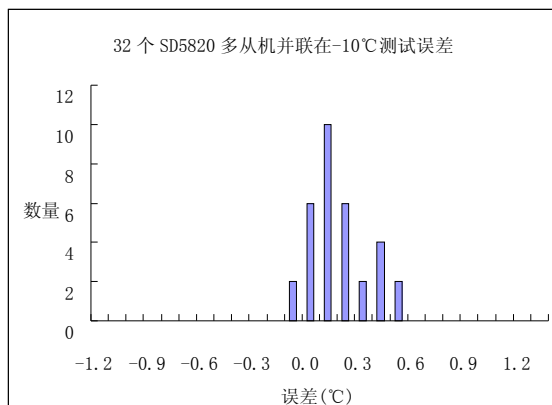


图 7. 32 个 SD5820 多从机并联在 -10°C 测试误差
单总线多点高精度测温解决方案 v0
<http://www.sdcmicro.cn>

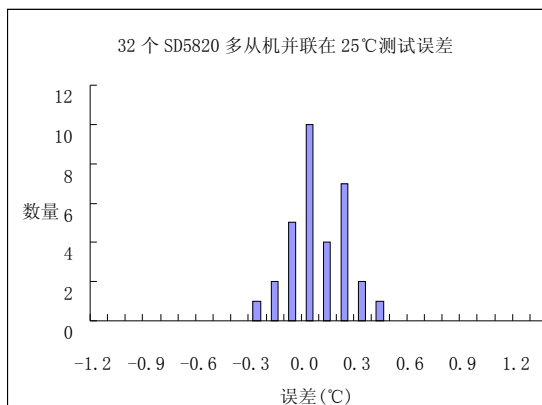


图 8. 32 个 SD5820 多从机并联在 25°C 测试误差
第 3 页 共 6 页
2017.6.7

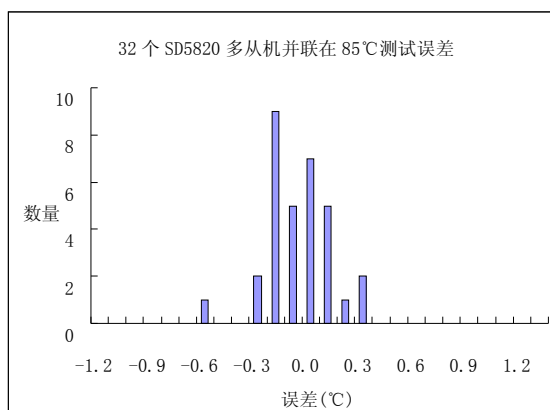


图 9. 32 个 SD5820 多从机并联在 85°C 测试误差

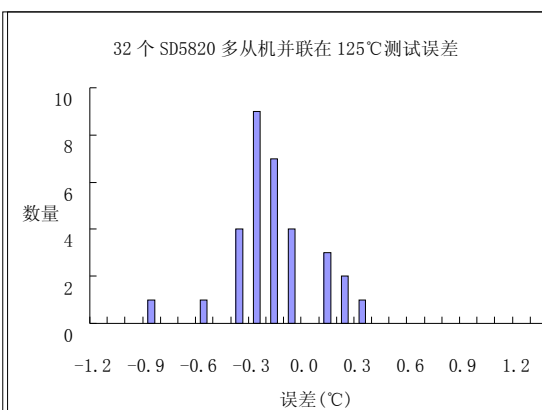


图 10. 32 个 SD5820 多从机并联在 125°C 测试误差

程序参考如下：

```
/******
```

```
* 杭州晶华微电子有限公司
```

```
* 2017.06.07
```

```
* Author:yanwangjun
```

```
* 编译器: AVR Studio with GCC
```

```
* Email:techsupport-4@sdicmicro.cn
```

```
*****
```

```
* SD5820 测试程序
```

```
*****
```

```
* 使用 Atmega16L 的 PD6 作为普通 I/O 读写 SD5820 芯片的温度数据, 使用 4M 晶振作为主时钟, GCC 编译器优化选项为 -O0 (没打开)。
```

```
*****
```

```
*modify:
```

```
*****/
```

```
#include<avr/io.h>
```

```
#include<avr/delay.h>
```

```
#include <stdarg.h>
```

```
// 使用 va_list 类型定义和相关宏定义
```

```
#define DIO_high DDRD &= ~(1 << DDD6) //配置 PD6 方向寄存器作为输入 (通过外部上拉成高电平)
```

```
#define DIO_low DDRD |= (1 << DDD6) //配置 PD6 方向寄存器作为输出 (通过在 main 函数中 PD6 被初始化为 0 时, 输出低电平)
```

```
#define DQ_PIN (PIND&0x40)
```

```
void delay_ms(unsigned int tt) //4M 时钟, 延时 ms
```

```
{
```

```
    while(tt--)
```

```
    {
```

```
        _delay_loop_2(1000);
```

```
    }
```

```
}
```

```
//复位, 握手
```

```
unsigned char Reset()
{
    unsigned char ack;
    DIO_high;
    delay_ms(1);    //先置高 1ms, 足够时间充电
    DIO_low;
    _delay_loop_2(800);
    DIO_high;
    _delay_loop_2(60);
    ack=DQ_PIN;
    _delay_loop_2(200);
    return(ack);
}
```

//读一个字节

```
unsigned char Read_byte()
{
    unsigned char i, data=0;
    for(i=0;i<8;i++)
    {
        data>>=1;
        DIO_low;
        _delay_loop_2(3);
        DIO_high;
        _delay_loop_2(3);
        if(DQ_PIN)
            data |=0x80;
        _delay_loop_2(60);
    }
    return(data);
}
```

//写一个字节

```
void Write_byte(unsigned char byte_data)
{
    unsigned char i, temp=0;
    for(i=0;i<8;i++)
    {
        temp=(byte_data&0x01);
        DIO_low;
        _delay_loop_2(4);
        if(temp)
            DIO_high;
        else
            DIO_low;
        _delay_loop_2(60);
    }
}
```

```

        DIO_high;
        _delay_loop_2(6);
        byte_data=byte_data>>1;
    }
}

int main()
{
    unsigned char temp_l=0,temp_h=0;
    PORTD &= ~(1 << PD6);           //初始化 PD6 为 0, 配合 DIO_high 和 DIO_low 语句使用。
    DIO_high;                       //设 PD6 为输入, 被外部上拉到高电平, DIO 空闲。

    delay_ms(5000);                 //延时 5S 等待寄生充电完才开始测试芯片

    Reset();
    Write_byte(0xCC);               //写跳过 ROM
    Write_byte(0x44);               //启动温度转换

    delay_ms(750);

    Reset();
    Write_byte(0xCC);
    Write_byte(0xBE);               //读寄存器指令

    temp_l=Read_byte();
    temp_h=Read_byte();             //读温度数据
    delay_ms(100);
    return 0;
}

```

修改记录:

版本号	修改日期	作者	修改记录
v0	2017-6-7	严玉军	初始版