

锐能微第三代单相计量芯片应用笔记

Data: 2014-3-20

Rev: 1.0

Data:2014.7.6

Rev: 1.1

Data:2014.12.22

Rev: 1.2

版本更新说明

版本号	修改时间	修改内容
V1.0	2014-3-20	创建
V1.1	2014.7.06	5.1 脉冲法校表步骤及算法： $HFConst = [23.2075 * Vu * Vi * 10^{11} / (EC * Un * Ib)]$ 修改为： $HFConst = [14.8528 * Vu * Vi * 10^{11} / (EC * Un * Ib)]$
V1.2	2014.12.22	5.1 4.相位校正计算公式规范 5.2 3.增益和相位单点校正 中公式更改： $\theta = [ArcSin(-ERR/1.732)] * (180/3.14159) / 0.02;$ 改为： $\theta = [ArcSin(-ERR/1.732)] / 0.02;$ 注： θ 为角度 $\theta = [ArcSin(-ERR/1.732)] * (180/3.14159) / 0.01;$ 改为： $\theta = [ArcSin(-ERR/1.732)] / 0.01;$ 注： θ 为角度
		5.1 1.确定基本参数 5.2 1.确定基本参数 6.1 确定基本参数 中 HFConst 计算公式更改 $HFConst = INT[14.8528 * Vu * Vi * 10^{11} / (EC * Un * Ib)]$ 改为： $HFConst = INT[16.1079 * Vu * Vi * 10^{11} / (EC * Un * Ib)]$
		“7 双路有功电能同时计量的实现” 中 EMUCON2 配置表格错，更改为： =00: 自定义电能输入选择为无功功率； =01: 自定义电能输入选择为通道 A 和通道 B 有功功率的矢量和； =10: 自定义电能输入选择为自定义功率寄存器 D2FP； =11: 自定义电能输入选择为通道 B 有功功率；
		SPI 写操作程序示例 c 程序 SPI 读操作程序示例 c 程序 更改错误，及增加备注
		5 校表方法 中 增加： 注意：有功 PF1.0 和 0.5L 校好后，无功 PF0.5L 的精度误差均值约-0.16%左右，需要再进行无功相位补偿

目录

1 概述	4
2 硬件电路设计.....	4
2.1 采样电路.....	5
2.2 基准电压电路.....	6
2.3 晶振电路.....	6
2.4 复位电路.....	7
2.5 芯片电源电路.....	8
2.6 SPI/UART通信接口电路.....	8
2.7 脉冲输出电路.....	9
3 可靠性设计.....	9
3.1 强电区域.....	10
3.2 电源和复位.....	11
3.3 通信接口.....	11
3.4 脉冲输出.....	12
3.5 晶体.....	12
4 软件设计.....	12
4.1 上电配置步骤.....	12
4.2 运行中计量芯片参数校验.....	13
4.3 SPI通信接口.....	13
5 校表方法.....	18
5.1 脉冲法校表步骤及算法.....	19
5.2 功率校表法步骤及算法.....	20
5.3 无功校正.....	22
5.4 有效值offset校正.....	22
5.5 启动功率设置.....	22
6 直流测量的应用.....	23
6.1 确定基本参数.....	23
6.2 直流offset校正.....	23
6.3 有效值OFFSET校正.....	24
6.4 电压、电流、功率转换系数确定.....	24
6.5 增益校正.....	24
7 双路有功电能同时计量的实现.....	25
8 应用注意事项.....	25

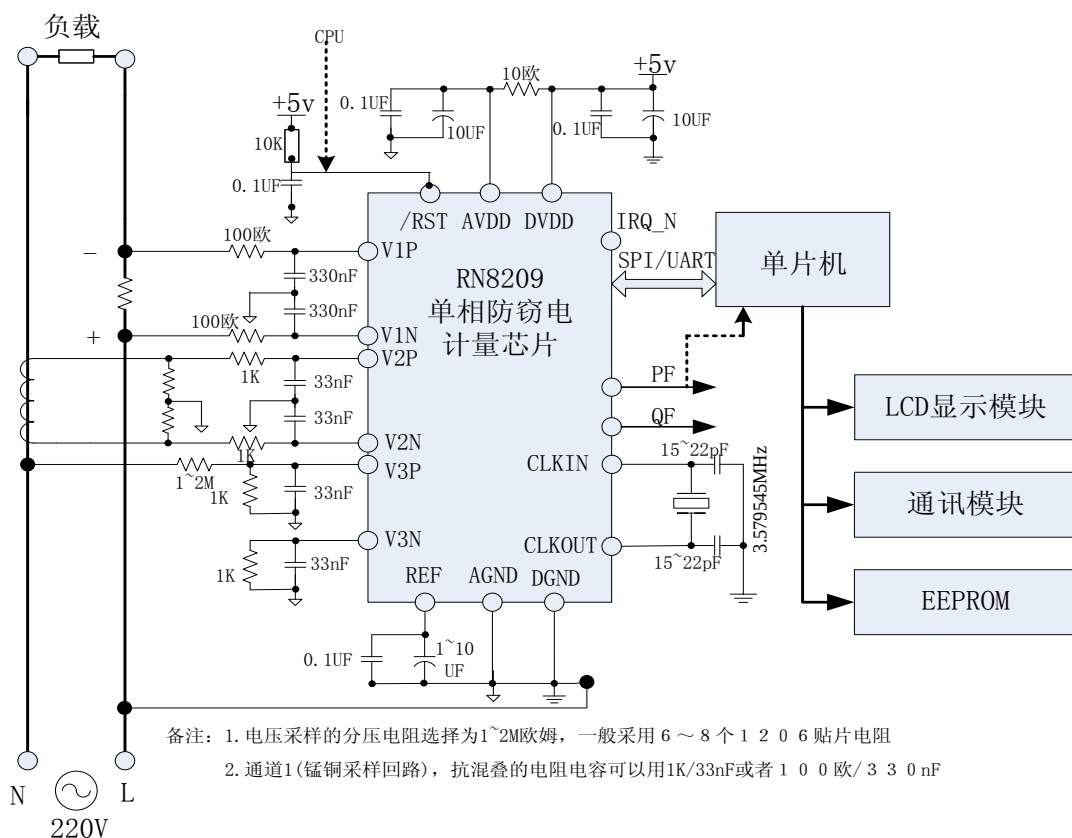
1 概述

本应用笔记介绍了使用 RN8209C、RN8209D 设计单相多功能电表的硬件设计、可靠性设计、软件设计方法和校表方法。阅读本文档时，请参阅相应的用户手册。

RN8209C、RN8209D 是一款两路电流输入，一路电压输入，支持采样零线电流的防窃电表的计量芯片，它可提供多种电力参数：有功电能、无功电能、自定义电能，有功功率、电流、电压、频率，其中有功功率和电流是同时提供火线和零线两路参数，用户可根据电流的大小进行电能计量通道的切换。

RN8207C 只是封装与 RN8209C/RN8209D 有所不同，具体使用方法可参阅用户手册及该应用笔记，不再单独描述。

2 硬件电路设计



用 RN8209D 设计的单相电能表参考原理框图

上图为 RN8209D 设计的单相电能表参考图。外围硬件包括电压电流采样电路、SPI/UART 通信接口电路、脉冲输出电路、电源和复位电路、时钟电路等。

RN8209C 与此设计图类似，cpu 与 RN8209C 仅需要通过 UART 口连接，外部复位引脚的功能与 RX 引脚复用。是否需要将有功脉冲输出口连接到 CPU 由客户决定。

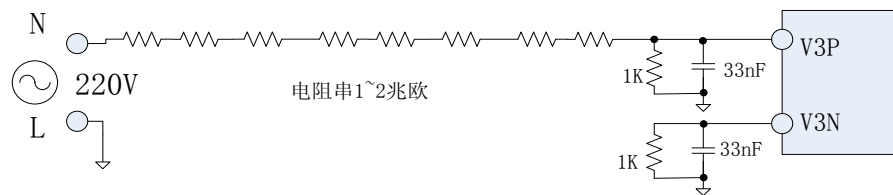
2.1 采样电路

RN8209 包含 3 路高精度 Σ - Δ ADC，三路 ADC 设计一致，ADC1（V1P/V1N 输入）为火线锰铜采样输入，ADC2（V2P/V2N 输入）为零线互感器采样输入，ADC3（V3P/V3N 输入）为电压采样输入。其中 ADC1 锰铜采样输入信号小，配置为 16 倍增益；ADC2 和 ADC3 输入信号大，配置为 1 倍增益。

电压采样电路：

推荐额定采样值 电压通道 ADC3 推荐输入信号为 100~220mVrms。

设计参考原理图：。

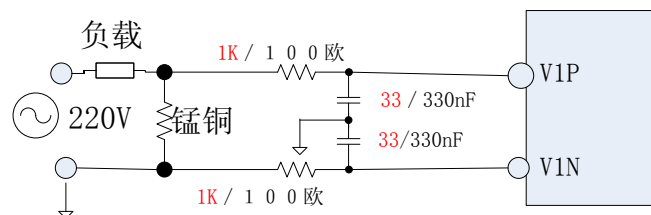


注意电阻分压串的电阻通常采用 6 个~8 个 1206 片阻，总阻值在 1 ~ 2 兆欧。

电压通道 ADC 增益配置成 1 倍：PGAU[1:0]=00。

抗混叠滤波电路推荐为 1K/33nF。

电流通道 A 采样电路：



通道 A 采用锰铜取样，其锰铜阻值根据电流规格进行选择，推荐：

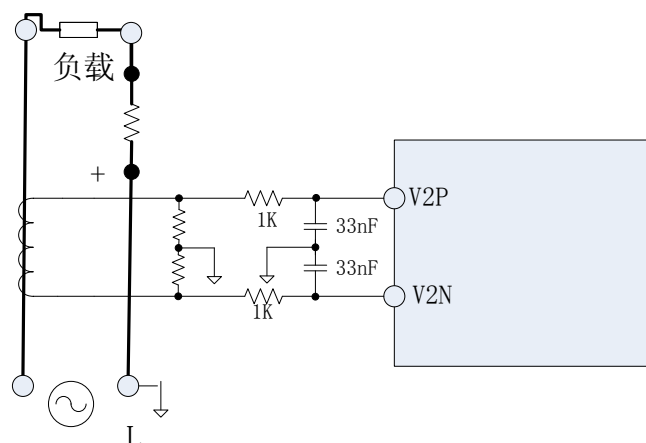
5 (6 0) A: 250~350 微欧

1 0 (1 0 0) A: 1 2 5 ~ 1 7 5 微欧

通道 ADC 增益配置成 1 6 倍：PGAIA[1:0]=11

抗混叠滤波电路可选择为 1K/33nF 或者 100 欧/330nF。

电流通道 B 采样电路：

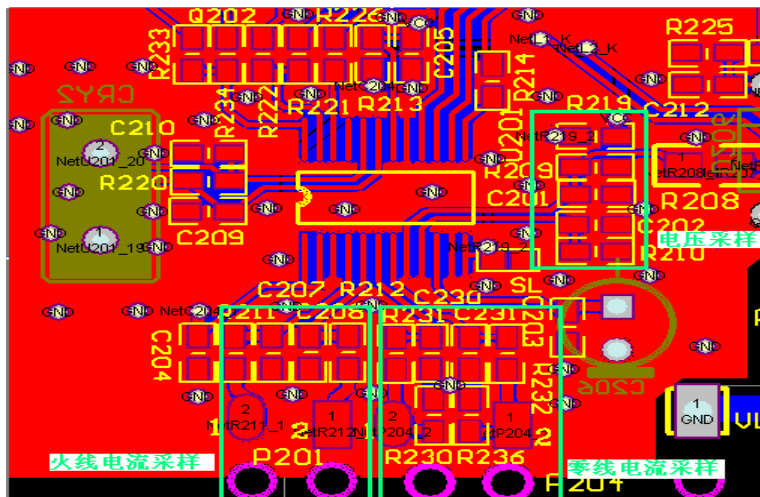


通道 B 采用互感器取样。抗混叠电路推荐为 1K/33nF。

通道 ADC 增益配置成 1 倍: PGAIB[1:0]=00。

Ib 点输入采样信号: 30 mv 左右。

采样电路 PCB 设计参考如下:



采样电路 PCB 设计参考图

采样电路 PCB 设计注意事项:

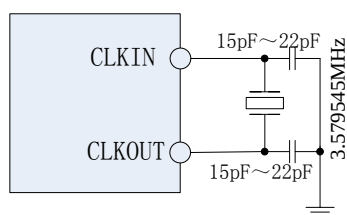
- 1、同一路 ADC 的 P 端和 N 端抗混叠滤波电容接地端尽量靠近；三路 ADC 的接地点需有大面积地平面，将三路 ADC 的接地点连接起来。
- 2、ADC 输入信号需走差分线。
- 3、ADC 输入信号应远离其他信号。

2.2 基准电压电路

RN8209 内置 $1.25V \pm 1\%$ 基准电压，电表设计时需在 RN8209 REF 引脚上加两个电容，其中一个为 $1 \sim 10\mu F$ (必须至少保证 $1\mu F$)，另外一个为 $0.1\mu F$ 电容。PCB 设计时，应注意基准电压走线尽量短，滤波电容尽量靠近芯片管脚。

2.3 晶振电路

RN8209 系统时钟为 3.579545MHz，请使用 3.579545MHz 晶振。参考设计电路请见图。

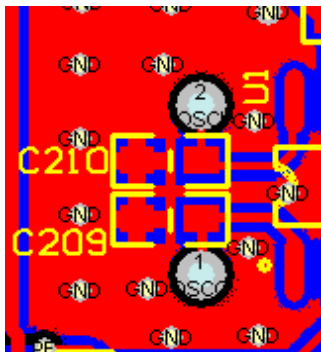


内部有 4M 欧姆串联电阻，外部不需要再加电阻。

推荐晶振参数：ESR<50，负载电容 10 ~ 15 pf。

考虑杂散电容影响,建议晶振外接两个 15~22pf 电容

pcb 设计注意：



1. 晶体紧靠管脚
2. 走线尽量短
3. 高频部分正反面铺地完整进行屏蔽
4. 电源及信号远离晶振电路

2.4 复位电路

芯片提供 3 种复位模式：

- 电源复位；
- 外部管脚复位（RN8209D、RN8209C 及 RN8207C 均支持 RX 管脚复位）；
- 命令复位；

其中命令复位与外部管脚复位等效，属于硬件复位。

外部复位管脚的处理（参见推荐电路）：上拉 10K 欧姆电阻，下接 0.1uf 电容到地；

任一全局复位发生时，寄存器恢复到复位初值，外部引脚电平恢复到初始状态。

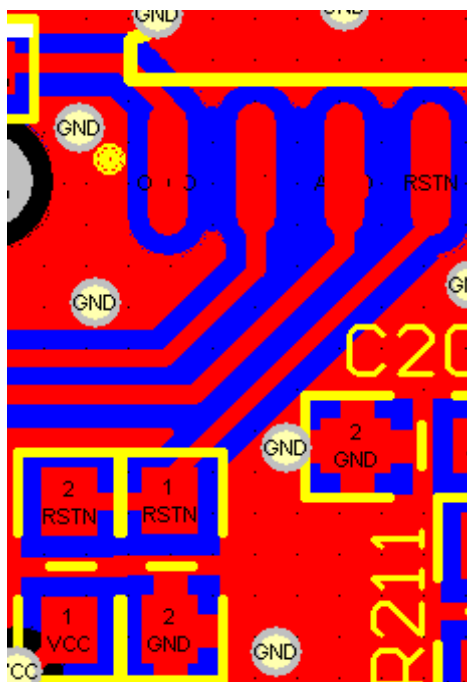
命令复位命令之后 15us，芯片完成复位。

RN8209D 的外部复位引脚高电平变低电平并保持 50us 以上，再由低电平变高电平 300us 以上后完成复位动作。

RN8209C 的 RX 引脚同时也是复位引脚，当输入信号低电平超过 20ms 时 RN8209C 认为是复位有效。此功能可在隔离应用情况下节省光耦数量。

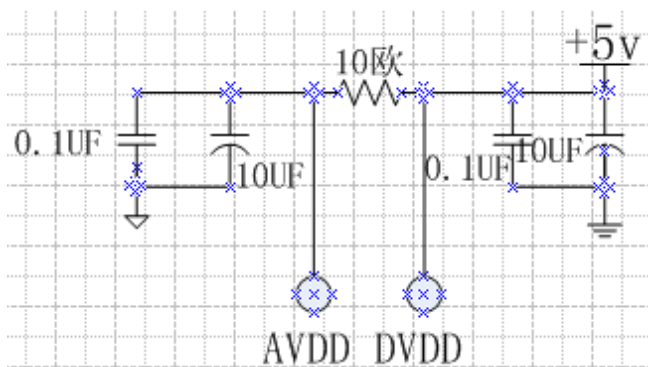
建议的 RX 引脚复位操作方式是：先将 RX 引脚置低 25ms，然后再将 RX 引脚置高 20ms，最后再开始正常的 UART 通信。

pcb 设计注意：



1. 电容靠近复位管脚
2. 复位管脚走线尽可能短
3. 铺地保护

2.5 芯片电源电路



- 说明：1.模拟电源与数字电源间接 10 欧电阻，减少数字噪声。
- 2.靠近 AVDD 和 DVDD 引脚处各接一个 0.1uf 去耦电容。PCB 设计时 0.1uf 电容紧靠管脚，电源走线通过电容后再到电源管脚
3. 电源电压范围是 3V-5.5V，选定典型供电电压（如 5V 或 3.3V）后，应保证电源波动在±10%范围内。

2.6 SPI/UART 通信接口电路

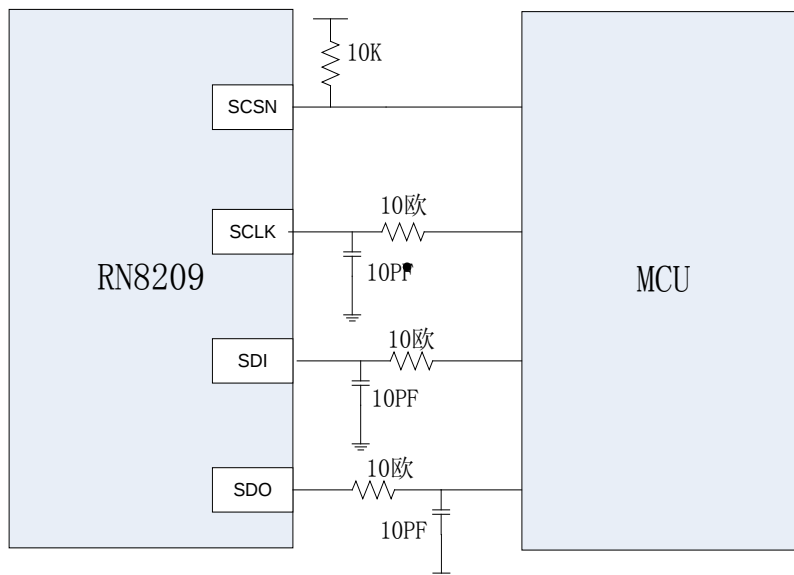
RN8209D 支持两种串行通信接口：SPI 和 UART。

当 IS 引脚接高电平时，选择为 SPI 通信接口。

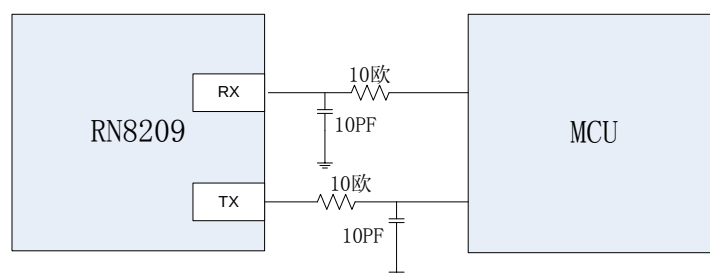
当 IS 引脚接低电平时，选择为 UART 通信接口，当需要光耦隔离时建议波特率 4800。

RN8209C 仅支持 UART，波特率固定为 4800。

传输信号线有可能受到干扰而出现抖动，为保证数据的可靠传输，需要外接 RC 进行滤波。参数的选择可根据需要确定。



SPI 接口参考电路原理图



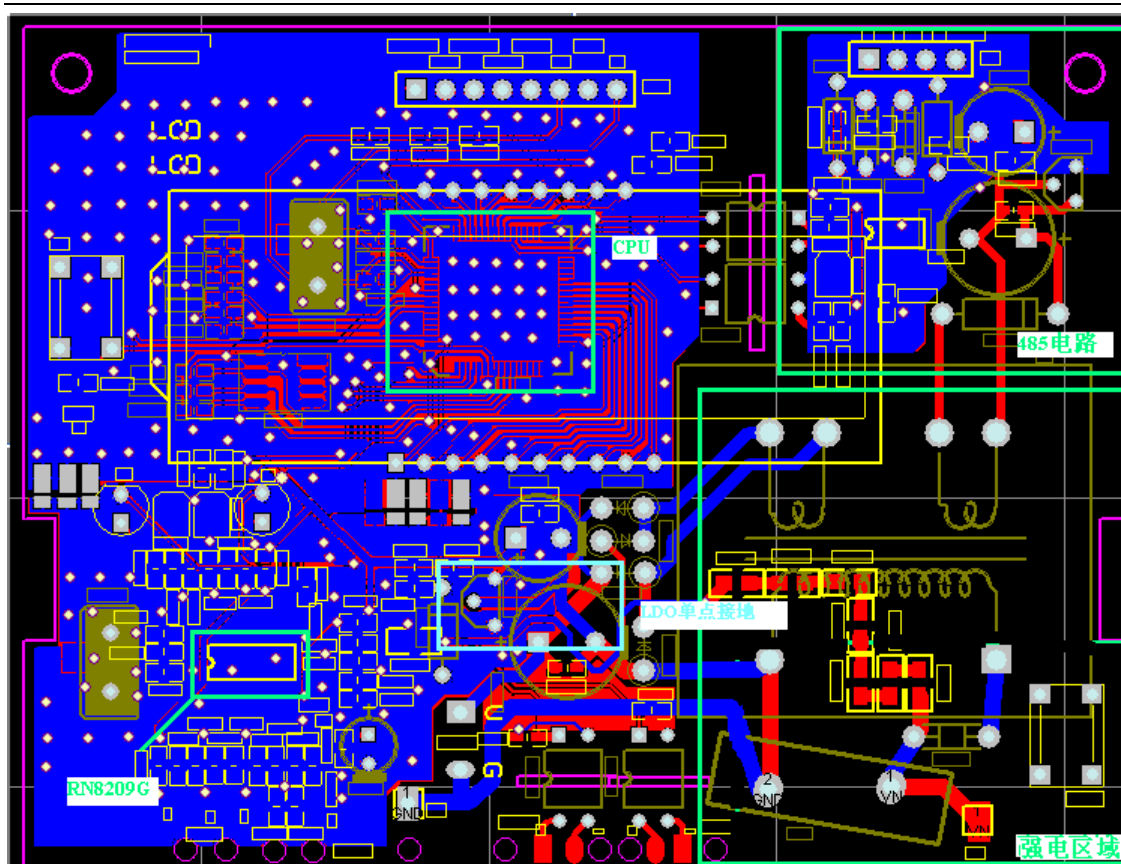
UART 接口参考电路原理图

SPI 和 UART 电路 PCB 设计时，应注意走线尽量短且远离其他信号线，并加地线作为屏蔽。

2.7 脉冲输出电路

RN8209 提供有功能量脉冲输出 PF 和自定义能量脉冲输出 QF，PF 和 QF 引脚输出驱动能力 5mA。推荐电路请参考图 1，其中限流电阻推荐使用 330 欧姆电阻，电阻过大可能导致输出信号幅度偏小，测试机台检测不到光耦输出信号。脉冲线一般走线较长，为消除噪声影响，需在光耦输入端附近加 2.2nf 去耦电容。

3 可靠性设计



用 RN8209 设计的单相电能表参考 PCB 图

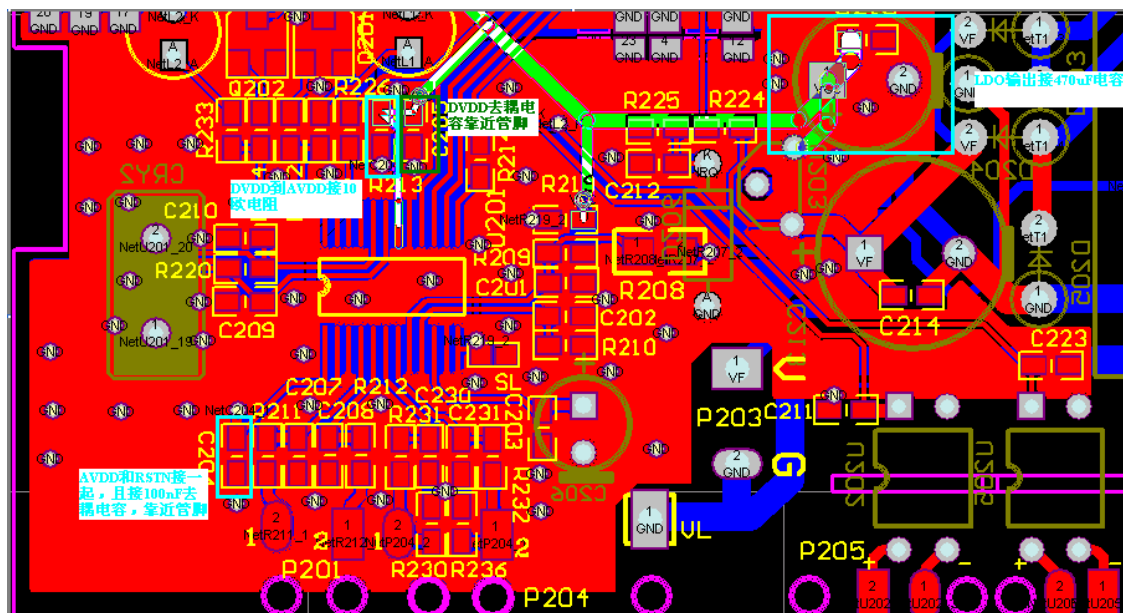
上图为 RN8209 设计的单相电能表 PCB 图，为使设计的电表满足电磁兼容性和可靠性，需遵循下列章节所属内容设计。

3.1 强电区域

安全及可靠性：参考上图。

- 1、电表设计时，需将强电和弱电隔离开；
- 2、PF 和 QF 脉冲输出需用光耦与 CPU 和计量部分隔离；
- 3、485 电路需用光耦与 CPU 和计量部分隔离。
- 4、电源与地之间加压敏电阻前端保护后到系统的电源
- 5、主电源线及地线要有足够的线距(建议 1.5mm 以上)

3.2 电源和复位



电源和复位参考 PCB 图

- 1、 建议 LDO 输出 DVDD 接 $\geq 470\mu\text{F}$ 的滤波电容。
- 2、 RN8209 数字电源 DVDD 和模拟电源 AVDD 之间需接 10 欧电阻,且在靠近 AVDD 和 DVDD 引脚处各接一个 100nf 去耦电容。
- 3、 外部复位管脚建议接 10k 上拉电阻和 100nf 去耦电容,且电容尽量靠近计量芯片管脚。
- 4、 计量芯片数字地和模拟地通过大面积铺地直接连接,不需要隔离。

3.3 通信接口

注意事项:

1. 通讯走线尽量短且远离其他信号线, 并加地线作为屏蔽。
2. 去耦电容尽量靠近管脚。

图 6 脉冲和晶振电路参考 PCB 图

脉冲输出限流电阻需靠近计量芯片；脉冲走线尽量远离其他信号线，并加地线屏蔽；去耦电容要靠近光耦输入端。

晶体外壳接地可增强抗干扰能力；晶体走线尽量短，且走线需加地线屏蔽；其他信号线应远离晶体电路。

4.1 上电配置步骤

上电后对 RN8209 进行参数配置遵循如下配置

1. **Mcu 对 RN8209 进行复位：**（可以是管脚复位，也可以是命令复位）

管脚复位:

RN8209D 高电平变低电平并保持 50us 以上，再由低电平变高电平 300us 以上后完成复位；RN8209C 先将 RX 引脚置低 25ms，然后再将 RX 引脚置高 20ms（RN8209D 也支持 RX 引脚复位）。

命令复位:

写使能（命令帧 0XEA 0XE5）状态下，复位命令下发（命令帧 0XEA 0XFA），有效接收正确后 300us 完成复位。

2. 校表参数配置 (须在写使能有效情况下):

HFCONST、增益、相位寄存器、系统控制字、计量控制字等 进行正确写入

3. 写禁止（在全部参数配置完成后）

4.2 运行中计量芯片参数校验

Mcu 须定时读出计量芯片的累加校验寄存器进行校验，如校验和改变，需重新复位并配置计量相关参数。

4.3 SPI 通信接口

SPI 接口信号说明

SCSN：SPI 从设备片选信号，低电平有效，输入信号，内部悬空，建议外接上拉电阻。

SCSN 由高电平变为低电平时，表示当前芯片被选中，处于通讯状态；SCSN 由低电平变为高电平时，表示通讯结束，通讯口复位处于空闲状态。

SCLK：串行时钟输入脚，决定数据移出或移入 SPI 口的传输速率。

所有的数据传输操作均与 SCLK 同步，RN8209 在上升沿将数据从 SDO 引脚输出；主机在上升沿将数据从 SDI 引脚输出。RN8209 和主机都在下降沿读取数据。

SDI：串行数据输入脚。用于把主设备数据传输到 RN8209 内部。

SDO：串行数据输出脚，用于把 RN8209 数据输出给主设备。SCSN 为高时，为高阻。

SPI 帧格式

SPI 帧包括读操作帧、写操作帧和特殊命令帧。每一帧的传输过程如下：

当 RN8209 检测到 SCSN 下降沿，SPI 进入通信方式，在此模式下，RN8209 等待 MCU 向命令寄存器传送命令字节。

命令寄存器是一个 8bit 宽的寄存器。对于读写操作，命令寄存器的 bit7 用来确定本次数据传输操作的类型是读操作还是写操作，命令寄存器的 bit6-0 是读写的寄存器的地址。对于特殊命令操作，命令寄存器的 bit7-0 固定为 0xEA H。

写完命令寄存器，芯片解析和响应命令，开始本次数据传输。数据传输结束后，SPI 又进入通信模式，等待 CPU 向命令寄存器传送新的命令字节。

这三种类型 SPI 帧格式说明见表 4-1。

表 4-1 SPI 帧格式

命令名称	命令寄存器	数据	描述
读命令	{0,REG_ADR[6:0]}	RDATA	从地址为 REG_ADR[6:0]的寄存器中读数据。 注意：读无效地址，返回值为 00h。
写命令	{1,REG_ADR[6:0]}	WDATA	向地址为 REG_ADR[6:0]的寄存器中写数据。
写使能命令	0xEA	0xE5	
写保护命令	0xEA	0xDC	
电流通道 A 选择命令	0xEA	0x5A	

电流通道 B 选择命令	0xEA	0xA5	
命令复位	0xEA	0xFA	复位命令，等效于外部 PIN 复位；当写使能之后，系统才接受该命令； 建议客户 CPU 对计量初始化前先进行软件复位或者 PIN 复位；

SPI 写操作

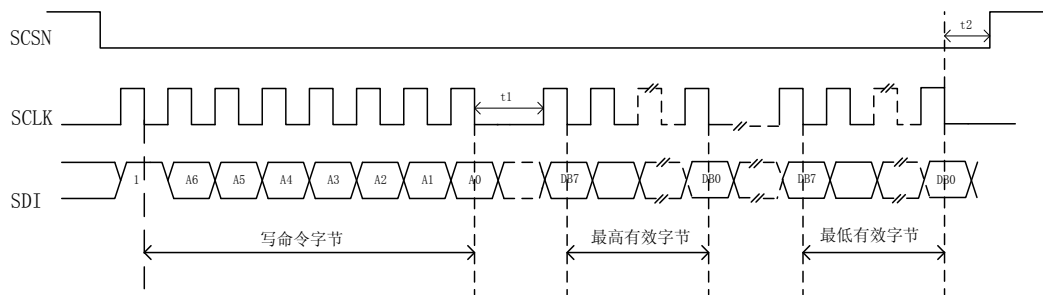


图 8 SPI 写时序

工作过程：

主机在 SCSN 有效后，先通过 SPI 写入命令字节（8bit，包含寄存器地址），再写入数据字节。注意：

1. 以字节为单位传输，高比特在前，低比特在后；
2. 多字节寄存器，先传输高字节内容，再传输低字节内容；
3. 主机在 SCLK 上升沿写数据，从机在 SCLK 下降沿取数据；
4. 数据字节之间的时间 t_1 要大于等于半个 SCLK 周期；
5. 最后一个字节的 LSB 传送完毕，SCSN 由低变高结束数据传输。SCLK 下降沿和 SCSN 上升沿之间的时间 t_2 要大于等于半个 SCLK 周期。

注意：有写保护功能的寄存器在写操作之前要先写入写使能命令。

SPI 写操作程序示例：汇编程序(使用瑞萨 78F0451 单片机):

```

TX_FRAME:                                ; 发送数据帧
    CLR1    SPI_CS                        ; 每次发送数据帧前，先将 SCSN 拉低
    MOV     A,T_COMM                      ; 发送命令字节
    CALL    !SEND_SPI_BYTE               ; 调用发送字节子程序
    NOP
SPIC1:   MOV     A,[HL]                   ; 发送数据预先存放在内存中，内存地址映射到 HL，发送时直接调用
    CALL    !SEND_SPI_BYTE               ; 调用发送字节子程序
    INCW    HL                           ; 装载下一个字节
    DBNZ    TD_NUM,$SPIC1                ; 根据发送字节数目，循环发送
    SET1    SPI_CS                        ; 发送数据帧完成后，SCSN 拉高
    RET
  
```

```
SEND_SPI_BYTE:                                ; 发送字节子程序
    MOV    B,#08H
SEND_SPI_BYTE_x:
    SET1    SPI_CL                ; SPI CLK 拉高
    ROLC    A,1                  ; 循环左移待发送数据
    MOV1    SPI_SO,CY             ; 每次循环将待发送数据的最高位输出到 SDO
    CLR1    SPI_CL                ; SPI CLK 拉低
    DBNZ    B,$SEND_SPI_BYTE_x    ; 循环 8 次，完成一个字节数据的发送。
    RET
```

SPI 写操作程序示例：c 程序

```
unsigned char TX_FRAME(unsigned char Num)
{
    unsigned char i;
    i=0;

    SPI_SCS=0;                        //片选打开
    for(;Num>0;Num--)
    {
        TX_ONEBYTE(TX_DATA[i]);      //发送地址及写数据

        i++;
    }

    NOP();
    .....;    //若干个 nop
    NOP();
    SPI_SCS=1;                        //片选禁止
    return(0);
}
```

```
void TX_ONEBYTE(unsigned char TxData)
{
    unsigned char i,j,k;
    i=0x80;
    j=8;
    for(;j>0;j--)
    {
        SPI_SCLK=1;
```

```

NOP();
.....;           //若干个 nop
NOP();
k=TxData & i;
if(k>0) SPI_SDO=1;
else    SPI_SDO=0;
NOP();
.....;           //若干个 nop
NOP();
SPI_SCLK=0;
i=i>>1;
NOP();
.....;           //若干个 nop
NOP();
}
}

```

SPI 读操作

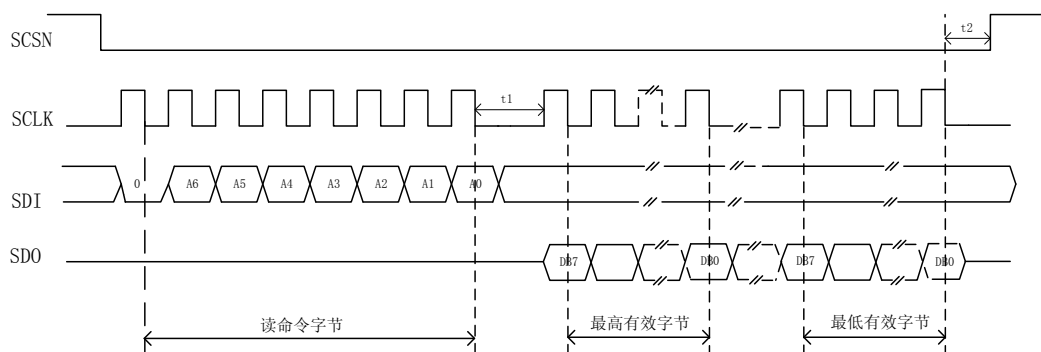


图 9 SPI 读时序

工作过程:

主机在 SCSN 有效后, 先通过 SPI 写入命令字节 (8bit, 包含寄存器地址), 从机收到读命令后, 在 SCLK 的上升沿将数据按位从 SDO 引脚输出。注意:

1. 以字节为单位传输, 高比特在前, 低比特在后;
2. 多字节寄存器, 先传输高字节内容, 再传输低字节内容;
3. 主机在 SCLK 上升沿写命令字节, 从机在 SCLK 上升沿将数据从 SDO 输出;
4. 数据字节的时间 t_1 要大于等于半个 SCLK 周期;
5. 最后一个字节的 LSB 传送完毕, SCSN 由低变高结束数据传输。SCLK 下降沿和 SCSN 上升沿之间的时间 t_2 要大于等于半个 SCLK 周期。

SPI 读操作程序示例汇编:

RX_FRAME: ; 接收数据帧

```

CLR1   SPI_CS           ; 每次接收数据帧前，先将 SCSN 拉低
MOV     A,T_ COMM       ; 发送命令字节
CALL    !SEND_SPI_BYTE  ; 调用发送字节子程序
SPIC3:  CALL    !RECEIVE_SPI_BYTE ; 调用接收字节子程序
MOV     [DE],A          ; 将接收到的字节放如内存
INCW    DE              ; 内存地址加 1
DBNZ    RD_NUM,$SPIC3   ; 根据接收字节数目，循环接收
SET1    SPI_CS          ; 每次接收数据帧完成后，将 SCSN 拉高
RET

RECEIVE_SPI_BYTE:       ; 接收字节子程序
MOV     A,#00H
MOV     B,#08H
RECEIVE_SPI_BYTE_x:
SET1    SPI_CL          ; SPI CLK 拉高
CLR1    SPI_CL          ; SPI CLK 拉低
MOV1    CY,SPI_SI       ; 将 SPI SDI 上的数据存放到进位标志中
ROL     A,1             ; 带进位循环左移，将 SDI 上的数据放到 A 的最低位
DBNZ    B,$RECEIVE_SPI_BYTE_x ; 循环接收 8 次，完成一个字节数据接收
RET

```

SPI 读操作程序示例 c 程序：

```

unsigned char RX_FRAME(unsigned char Num)
{
    unsigned char i;
    i=0;

    SPI_SCS=0;           //片选打开
    TX_ONEBYTE(TX_DATA[0]); //发送读地址

    for(;Num>0;Num--)
    {
        RX_DATA[i]=RX_ONEBYTE(); //接收数据

        i++;
    }

    NOP();
    .....;    ///若干个 nop
    NOP();
    SPI_SCS=1;           //片选禁止

```

```
        return(0);
    }

    unsigned char RX_ONEBYTE(void)
    {
        unsigned char i,j;
        i=0x00;
        j=8;
        for(;j>0;j--)
        {
            i=i<<1;
            SPI_SCLK=1;
            NOP();
            .....;          //若干个 nop
            NOP();
            SPI_SCLK=0;
            NOP();
            NOP();
            NOP();
            if(SPI_SDI==0) i=i|0x00;
            else          i=i|0x01;
            NOP();
            .....;          //若干个 nop
            NOP();
        }
        return(i);
    }
}
```

5 校表方法

设计一块 220v (U_n)、5A (I_b) 额定输入、表常数为 3200 (EC) 的电表。A 通道电流采样使用 350 微欧的锰铜，通道 A 模拟通道增益为 16 倍；B 通道电流采样使用互感器，选择通道 B 模拟增益为 1 倍；电压采用电阻分压输入，模拟通道增益为 1 倍，芯片引脚上采样电压值为 0.22v。

注意：有功 PF1.0 和 0.5L 校好后，无功 PF0.5L 的精度误差均值约-0.16%左右，需要再进行无功相位补偿

5.1 脉冲法校表步骤及算法

1.确定基本参数：hfconst，校表参数清为默认值

根据硬件设计确认电压、电流 ADC 输入信号（需乘以 ADC 增益倍数，单位 V），计算合适的 HFConst 值：

$$\text{HFConst} = [16.1079 * V_u * V_i * 10^{11} / (EC * U_n * I_b)]$$

V_u : 电压通道输入值，220V 分压后得到，一般选择为 0.1~0.22V；

V_i : 电流通道取样值，需要乘以增益倍数，如 $5A * (350 \text{ 微欧} / 10^6) * 16 = 0.028V$ ；

EC: 电表脉冲常数

U_n : 额定电压 单位：V，

I_b : 额定电流 单位：A

2.确定电压、电流、功率转换系数：

表台加 U_n I_b 读出计量芯片电压有效值 V、电流有效值 I，计算：

$K_v = U_n / V$; 电压转换系数，该系数与寄存器测量值相乘即得到输入的电压 (v)

$K_i = I_b / I$; 电流转换系数，该系数与寄存器测量值相乘即得到输入的电流(A)

$K_p = 3.22155 * 10^{12} / (2^{32} * \text{HFConst} * EC)$; EC 脉冲常数

; 功率转换系数，该系数与寄存器测量值相乘即得到输入的功率(w)

3.增益校正：

台体加 U_n 、 I_b 、1.0，读误差 ERR，校正值计算：

$$\text{PGAIN} = -\text{ERR} / (1 + \text{ERR})$$

如果 $\text{PGAIN} > 0$ ，校正值是 $\text{PGAIN} * 2^{15}$;

如果 $\text{PGAIN} < 0$ ，校正值是 $\text{PGAIN} * 2^{15} + 2^{16}$;

4.相位校正

台体加 U_n 、 I_b 、0.5L，读误差 ERR，校正值计算公式如下：

$$\theta = [\text{ArcSin} (-\text{ERR} / 1.732)] / 0.02; \quad \text{注：} \theta \text{ 为角度}$$

如果 $\theta > 0$ ，校正值是将 θ 取整；

如果 $\theta < 0$ ，校正值是将 $\theta + 2^8$ 后取整

写入角度校正寄存器 PHSA

若提高相位校正的准确度（1bit 对应 0.01 度）：则计算公式：

$$\theta = [\text{ArcSin} (-\text{ERR} / 1.732)] / 0.01; \quad \text{注：} \theta \text{ 为角度}$$

如果 $\theta > 0$ ，校正值是将 θ 取整；

如果 $\theta < 0$ ，校正值是 $\theta + 2^9$ 后取整

将最低位写入 PhsA0 (EMUCON2 的 Bit8)，高 8 位写入 PHSA

5.有功偏置 Offset 的校正方法:

A. 5%Ib 点的功率值做为校正依据：表台加 5%Ib 电流 Un，读出计量芯片的功率寄存器值，求至少 20 次平均值 P，与标准表的功率值 P0，计算功率 offset 值：

$$APOSA = [P0 * (1/Kp) - P] / (1 + GPQA)$$

如果 $APOSA < 0$ ，则 $APOSA + 2^{16}$

P: 芯片测量值平均值

P0: 标准表显示功率

Kp: 功率转换系数

GPQA: 功率增益归一化值

APOSA : 功率 offset 值

B. 5%Ib 点的电能误差做校正的依据:

$$APOSA = (P0 * 1/Kp) * (-err) / (1 + GPQA) \quad (err < 0 \text{ 时})$$

$$= 2^{16} + (P0 * 1/Kp) * (-err) / (1 + GPQA) \quad (err > 0 \text{ 时})$$

APOSA : 功率 offset 值

P0: 标准表显示功率 err 标准表显示误差

Kp: 功率转换系数

GPQA: 功率增益归一化值

5.2 功率校表法步骤及算法

1.确定基本参数：合适的 hfconst 值，校表参数清为默认值

Hfconst 确认公式：

$$HFconst = \text{INT}[(16.1079 * Vu * Vi * 10^{11}) / (Un * Ib * Ec)]$$

Vu: 电压采样信号，220V 分压后得到，一般选择为 0.1~0.22v 左右；

Vi: 电流采样值，需要乘以增益倍数，如 $5A * 350 \text{ 微欧} / 10^6 * 16 = 0.028v$ ；

EC: 电表脉冲常数

Un: 额定电压 220V

Ib: 额定电流 5A

2.电压、电流、功率转换系数确定:

表台加 Un Ib 读出计量芯片电压有效值测量值 V、电流有效值测量值 I，计算：

$Kv = Un/V$;电压转换系数，该系数与寄存器测量值相乘即得到输入的电压 (v)

$K_i = I_b / I$; 电流转换系数, 该系数与寄存器测量值相乘即得到输入的电流(A)
 $K_p = 3.22155 \times 10^{12} / (2^{32} \times \text{HFConst} \times \text{EC})$; EC 脉冲常数
; 功率转换系数, 该系数与寄存器测量值相乘即得到输入的功率(w)

3.增益和相位单点校正:

台体加 UN、IB、0.5L, 读出电压 (U 测)、电流 (I 测)、有功功率寄存器值 (P 测), 根据视在功率偏差进行增益校正; 根据有功功率偏差及计算出的增益校正值进行相位校正。

增益校正公式:

$$\text{PGAIN} = -\text{ERR} / (1 + \text{ERR})$$

如果 $\text{PGAIN} > 0$, 校正值是 $\text{PGAIN} \times 2^{15}$;

如果 $\text{PGAIN} < 0$, 校正值是 $\text{PGAIN} \times 2^{15} + 2^{16}$;

ERR 计算公式:

$$\text{ERR} = (\text{S 测量} - \text{S 标准}) / \text{S 标准}$$

$$\text{S 测量} = \text{U 测} \times \text{I 测} = (\text{Ureg} / 2^{23}) \times (\text{Ireg} / 2^{23})$$

$$\text{S 标准} = \text{U 标准} \times \text{I 标准} \times 1 / K_p / 2^{31} \quad (\text{电压 } 0.001\text{V} \quad \text{电流 } 0.0001\text{A})$$

Ureg: 电压有效值寄存器值

Ireg: 电流有效值寄存器值

U 标准: 标准表显示电压有效值 单位 V 3 位小数

I 标准: 标准表显示电流有效值 单位 A 4 位小数

相位校正公式:

$$\theta = [\text{ArcSin}(-\text{ERR} / 1.732)] / 0.02 \quad ; \quad \theta \text{ 为角度}$$

$\theta > 0$, 校正值是将 θ 取整;

$\theta < 0$, 校正值是 $\theta + 2^8$ 后取整

ERR 计算公式:

$$\text{ERR} = [\text{P 测} \times (1 + \text{Pgain 归一化}) - \text{P 标准}] / \text{P 标准}$$

$$\text{P 标准} = \text{P0} \times 1 / K_p$$

P 测量: 与电压、电流 一同读出来的有功功率寄存器值

P0: 标准表显示功率值 单位 w 4 位小数位 0.0001w

Pgain 归一化: 增益校正后的有功增益寄存器的归一化值, 公式:

$$\text{Pgain 归一化} = \text{Pgain} / 2^{15} \quad ; \quad \text{Pgain 寄存器最高位} = 0$$

$$\text{Pgain 归一化} = (\text{Pgain} - 2^{16}) / 2^{15} \quad ; \quad \text{Pgain 寄存器最高位} = 1$$

若提高相位校正的准确度 (1bit 对应 0.01 度): 则公式:

$$\theta = [\text{ArcSin}(-\text{ERR} / 1.732)] / 0.01 \quad ; \quad \theta \text{ 为角度}$$

如果 $\theta > 0$, 校正值是将 θ 取整;

如果 $\theta < 0$, 校正值是 $\theta + 2^9$ 后取整

将最低位写入 Phsx0 (EMUCON2 的 Bit8/bit9), 高 8 位写入 PHSx

4.有功偏置 Offset 的校正方法:

5%Ib 点的功率值做为校正依据:

1) 表台加 5%Ib 电流 Un, 读出计量芯片的功率寄存器值, 求至少 20 次平均得 P, 与标准表的功率值 P0, 计算功率 offset 值:

$$APOSa = [P0 * (1/Kp) - P] / (1 + GPQA)$$

P: 芯片寄存器测量值平均值

P0: 标准表显示功率

Kp: 功率转换系数

GPQA: 功率增益归一化值

结果>0, 直接写入 APOSa 寄存器

结果<0, +2¹⁶ 后写入 APOSa 寄存器

B 通道有功校正和 A 通道类似。

在做功率法校表时, 可以读取多点的有效值、功率做平均, 以提高校正精度。

5.3 无功校正

注意: 无功校正必须在有功校正后进行。

无功相位补偿寄存器计算方法如下:

若标准表在 A 通道、Imax、PF=0.5L(30°)上读出误差为 err, 则:

$$\alpha = \text{err} / \cot(\theta) = \text{err} * 0.5774。$$

如果 $\alpha \geq 0$, 则 $Q_{\text{phs}} = \text{INT}[\alpha * 2^{15}]$; 如果 $\alpha < 0$, 则 $Q_{\text{phs}} = \text{INT}[2^{16} + \alpha * 2^{15}]$

无功 Offset 校正方法参照有功 Offset 校正。

5.4 有效值 offset 校正

芯片提供了电流有效值偏置校正寄存器, 在电流输入为零的条件下, 读取电流有效值寄存器的值为 0x000483, (可以读若干次取平均值) 十进制数为 1155。将其平方后求其反码: $1155 * 1155 = 1334025 = 0x145B09$, 32 位反码为 0Xffeba4f7。取中间 4 位数 0xeba4 写入电流有效值偏置校正寄存器。公式:

$$IARMSOS = \sim [(IARM_{\text{多次平均}})^2 / 2^{16}] \text{ 低 16 位}$$

B 通道与 A 类似。

5.5 启动功率设置

电表需要配置合适的有功、无功启动功率, 确保正常启动和防潜。设置的依据 50 ~ 80 % 启动功率进行设置, 公式:

$$P_{\text{Start}} = 80 \% * P_{\text{启动}} * (1/Kp) / 2^8 ; \text{式中是按 } 80 \% \text{ 启动功率计算}$$

无功启动功率设置同有功启动类似。

6 直流测量的应用

使用 RN8209 能够实现直流测量，与交流测量的不同点在于需要增加直流偏置校正，校正步骤如下：

6.1 确定基本参数

HFconst 确认公式：

$$\text{HFconst} = \text{INT}[(16.1079 * V_u * V_i * 10^{11}) / (U_n * I_b * E_c)]$$

V_u ：额定电压采样信号，一般选择为 0.1~0.22v 左右；

V_i ：额定电流采样值，一般选择为 0.075V

E_c ：电表脉冲常数

U_n ：额定电压 V，

I_b ：额定电流 A

其他校表参数清为默认值。

6.2 直流 offset 校正

步骤 1：寄存器配置：

SYSCON 中 BIT[1:0]、BIT[3:2]、BIT[5:4]写入 0，配置三路 ADC 为 1 倍增益；

EMUCON 中将 IA/IB/U 三路 ADC 的高通使能关闭，BIT[14]、BIT[6:5]配置为 1；

步骤 2：求有效值平均值

输入接地，读 IA、IB、U 三路的有效值 10 次，分别计算有效值的平均值：

IARMS 平均值

IBRMS 平均值

URMS 平均值

步骤 3：直流 OFFSET 校正

由于有效值读出来为正，而直流 OFFSET 可能为正，也可能为负，因此写入直流 OFFSET 后需要有一验证过程：写入后稳定 2s 后读出有效值与写入前的有效值进行比较，如写入后的值变小趋向 0，则校正完成；如写入后的值变大趋向 2 倍，则前面写入值取反后重新写入。如下：

1 第一次写入

IARMS1 平均值的 BIT[23:8]先写入 DCIAH 寄存器, BIT[7:4]后写入 DCL 寄存器的 BIT[3:0]

IBRMS1 平均值的 BIT[23:8]写入 DCIBH 寄存器, BIT[7:4]写入 DCL 寄存器的 BIT[7:4]

URMS1 平均值的 BIT[23:8]写入 DCUH 寄存器, BIT[7:4]写入 DCL 寄存器的 BIT[11:8]

2 验证

等待 2S 后读 IA、IB、U 三路的有效值 10 次, 计算有效值的平均值 IARMS2、IBRMS2、URMS2, 若有效值相对于未校正前变小, 则校正完成, 若值变大为原来的约为 2 倍, 则需继续进行下一步操作;

3 取反重新写入

~IARMS1 平均值的 BIT[23:8]写入 DCIAH 寄存器, BIT[7:4]写入 DCL 寄存器的 BIT[3:0],

~IBRMS1 平均值的 BIT[23:8]写入 DCIBH 寄存器, BIT[7:4]写入 DCL 寄存器的 BIT[7:4],

~URMS1 平均值的 BIT[23:8]写入 DCUH 寄存器, BIT[7:4]写入 DCL 寄存器的 BIT[11:8];

6.3 有效值 OFFSET 校正

在直流 OFFSET 校正完成后, 稳定 2s 中, 读 IA、IB 电流的有效值 10 次, 求各自平均值进行有效值 offset 校正, 公式如下:

$$IARMSOS = \sim \left[\left(IARMS_{\text{多次平均}} \right)^2 / 2^8 \right]_{\text{低16位}}$$

$$IBRMSOS = \sim \left[\left(IBRMS_{\text{多次平均}} \right)^2 / 2^8 \right]_{\text{低16位}}$$

6.4 电压、电流、功率转换系数确定

表台加 U_n I_b 读出计量芯片电压有效值测量值 V 、电流有效值测量值 I , 计算:

$K_v = U_n / V$; 电压转换系数, 该系数与寄存器测量值相乘即得到输入的电压 (v)

$K_i = I_b / I$; 电流转换系数, 该系数与寄存器测量值相乘即得到输入的电流(A)

$K_p = 3.22155 \times 10^{12} / (2^{32} \times HFConst \times EC)$; EC 脉冲常数

; 功率转换系数, 该系数与寄存器测量值相乘即得到输入的功率(w)

6.5 增益校正

台体加 U_N 、 I_B , 读出有功功率值 (可读多点平均) $P_{\text{测}}$, 按校正公式计算:

$$ERR = (P_{\text{测}} \times K_P - U_n \times I_b) / (U_n \times I_b)$$

$$PGAIN = -ERR / (1 + ERR)$$

如果 $PGAIN > 0$, 校正值是 $PGAIN \times 2^{15}$;

如果 $PGAIN < 0$, 校正值是 $PGAIN \times 2^{15} + 2^{16}$;

7 双路有功电能同时计量的实现

通过配置 EMUCON2 的

5,4	D2FM[1:0]	=00: 自定义电能输入选择为无功功率; =01: 自定义电能输入选择为通道 A 和通道 B 有功功率的矢量和 =10: 自定义电能输入选择为自定义功率寄存器 D2FP; =11: 自定义电能输入选择为通道 B 有功功率;
-----	-----------	--

进行自定义计量, 如需要双路同时计量, 配置 D2FM[1:0]=01, 这样 B 通道有功电能的计算结果存在 EnergyD 和 EnergyD2, 脉冲输出是 QF 脚。

8 应用注意事项

- 1、注意通道切换是特殊命令字, 不反映在校验和中, 通过 EMU 状态标志位查询。
- 2、电表设计时需在 RN8209 REF 引脚上加两个电容, 其中一个为 1~10uF(必须至少保证 1uF), 另外一个为 0.1uF 电容。
- 3、晶振电路内部已有 4M 欧姆串接电阻, 外部电路不需要再加电阻。