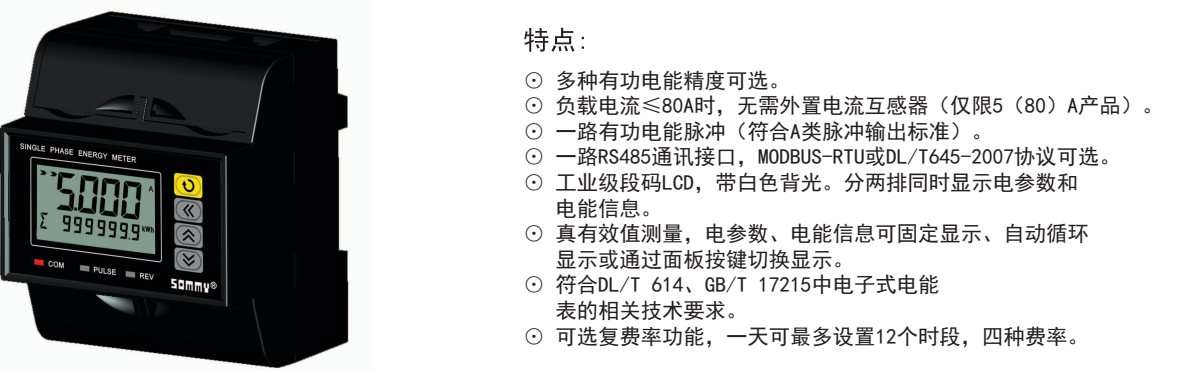


GQ7系列导轨安装式单相电能表操作说明书



- 特点:
1. 多种有功电能精度可选。
2. 负载电流≤80A时,无需外置电流互感器(仅15(80)A产品)。
3. 一路有功电能脉冲(符合A类脉冲输出标准)。
4. 一路RS485通讯接口,MODBUS-RTU或DL/T645-2007协议可选。
5. 工业级段码LCD,带白色背光,两排同时显示电参数和电能信息。
6. 具有有效测量、电参数、电能信息可固定显示、自动循环显示或通过面板按键切换显示。
7. 符合DL/T 614、GB/T 17215中电子式电能表的相关技术要求。
8. 可选费率功能,一天可最多设置12个时段,四种费率。

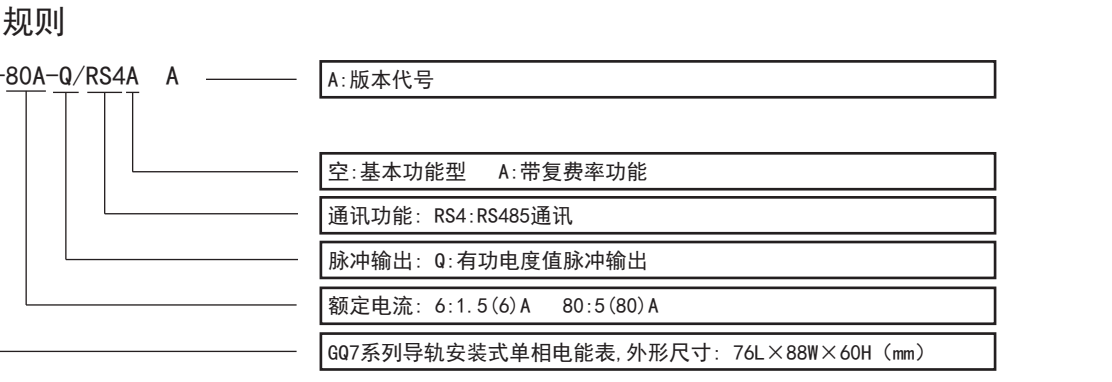


Table with 7 columns: Model, Accuracy, Rated Voltage, Current Specification, Connection Type, Rate Function, Communication Function, Pulse Constant.

面板说明

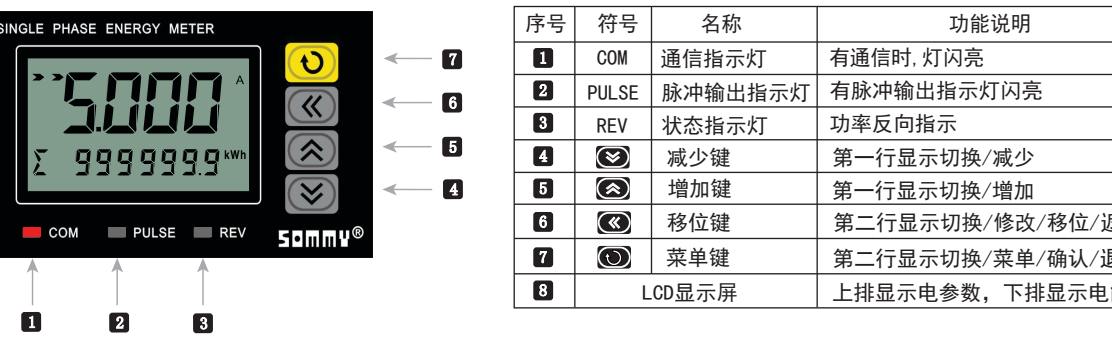
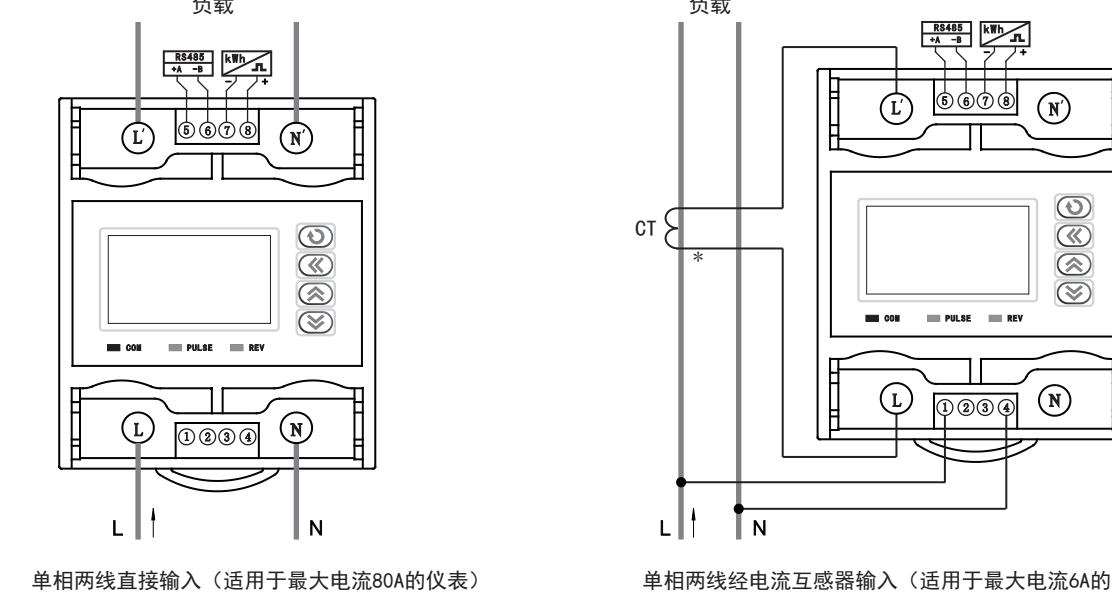


Table with 4 columns: No., Symbol, Name, Function Description. Lists indicators like COM, PULSE, REV, and buttons like SEL, CLR, etc.

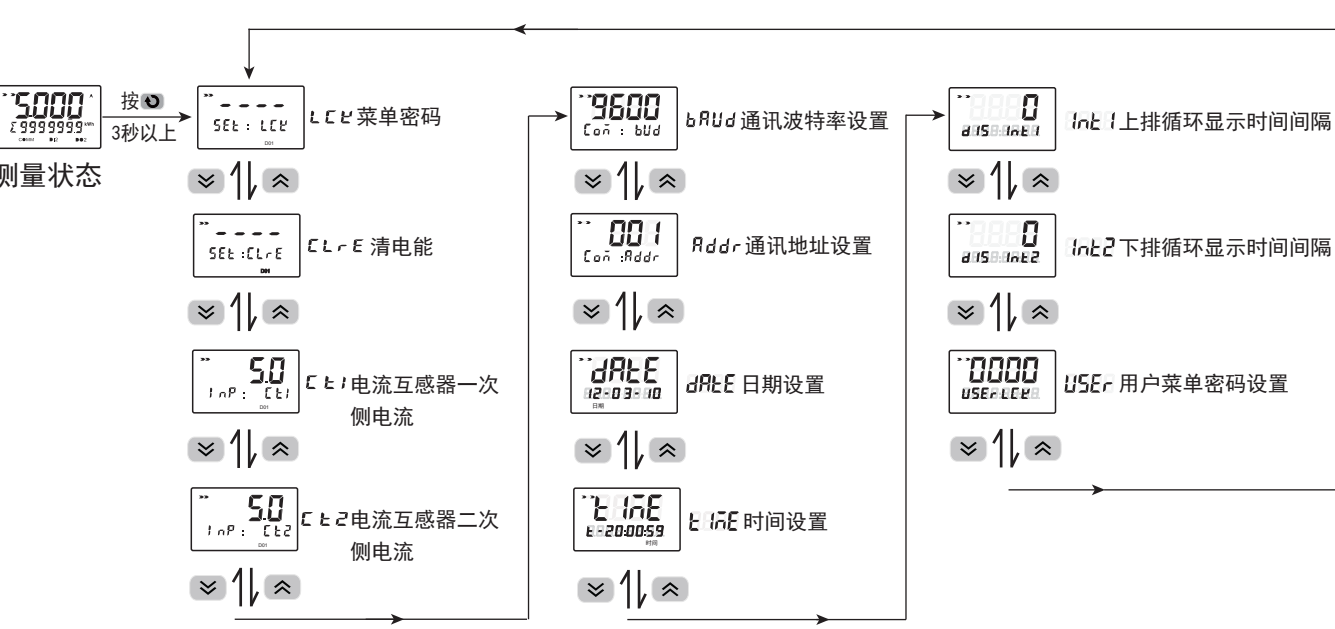
接线图



操作说明

- 1. 测量状态下,分别按“<=>”和“<=>”切换循环显示累计有功电量,无功电量,累计尖电量,累计峰电量,累计平电量,累计谷电量...
2. 在测量状态下,按“<=>”、“<=>”切换循环显示电压有效值,电流有效值,有功功率值,无功功率值,功率因数,电网频率...
3. 进入菜单,按“<=>”约3秒进入设置菜单,按“<=>”向前循环各菜单,按“<=>”向后循环各菜单...
4. 修改菜单内参数设置:进入菜单后,按“<=>”或“<=>”找到需要修改的菜单项,按“<=>”第二排参数值开始闪动...
5. 进入复费率菜单,在测量状态下,长按“<=>”键3秒进入复费率设置菜单,短按“<=>”或“<=>”键切换菜单...
6. 退出菜单:在菜单状态下按“<=>”键约3秒可退出菜单到正常测量状态...

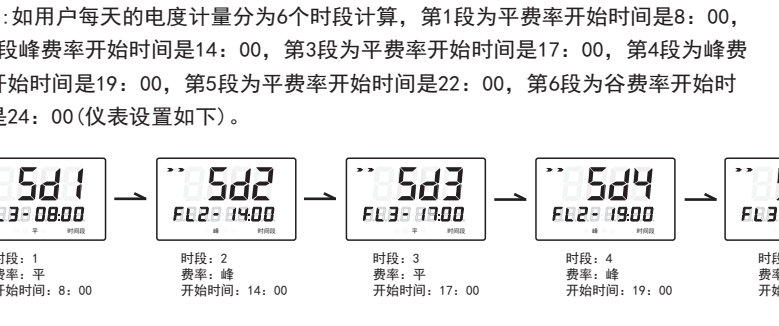
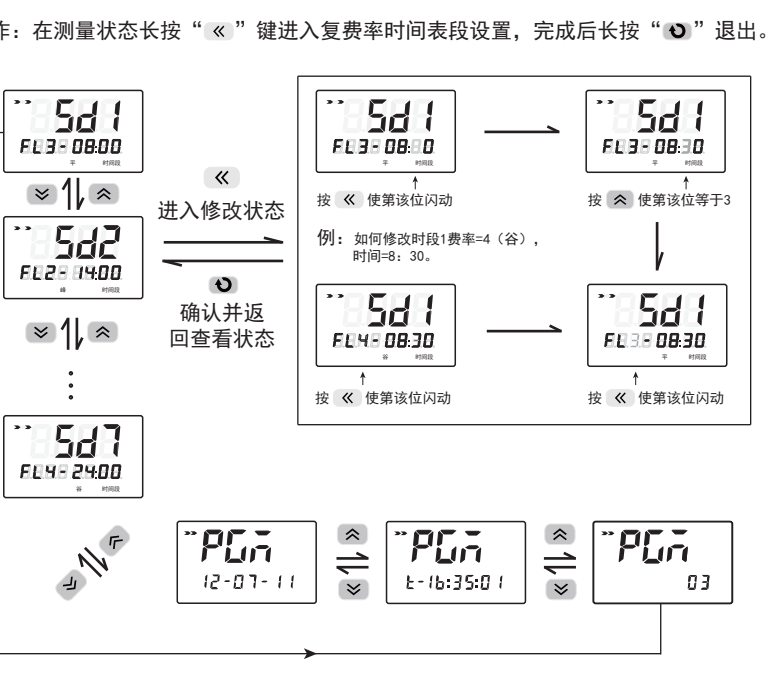
菜单操作流程



菜单结构

Table with 5 columns: No., Display Character, Represented Parameter, Parameter Range, Description, Default Value. Lists menu items like user code, clear energy, CT current, communication rate, address, date, time, and interval settings.

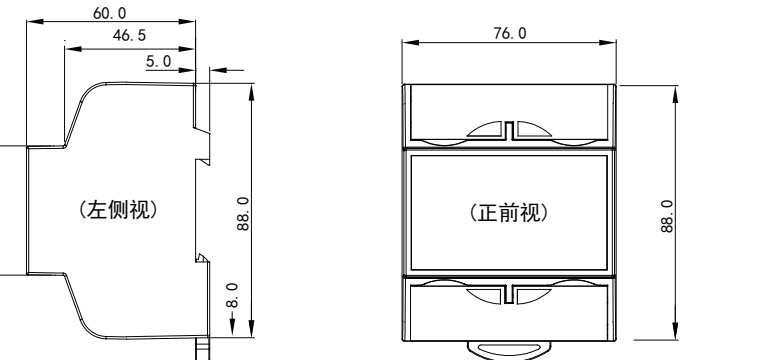
复费率菜单操作说明



参数说明

Table with 3 columns: Parameter Name, Description, Unit/Value. Explains terms like cumulative energy, peak energy, valley energy, and interval settings.

外形尺寸及安装图



通信协议

- 007系列表使用Modbus RTU通信协议,进行RS485半双工通信。读功能号0x03,写功能号0x06,采用16位CRC校验,仪表对校验错误不返回。
通信异常处理:异常应答时,将功能号的最高位置1。例如:主机请求功能号0x04,则从机返回的功能号对应项为0x04。
错误应答码:
0x01—功能码非法:仪表不支持接收到的功能号。
0x02—数据位置非法:主机指定的数据位置超出仪表的范围。
0x03—数据值非法:主机发送的数据值超出仪表对应的数据范围。

Tables for Modbus communication: Host request (读多寄存器), Slave response (读多寄存器), and Slave response (写多寄存器).

功能号异常应答

Table for Modbus slave response when the function code is 0x04, showing error codes and their meanings.

写多寄存器

Table for Modbus slave response when writing multiple registers, showing error codes and their meanings.

从机正常应答

Table for Modbus slave response when the function code is 0x04, showing data format and error codes.

数据位置错误应答

Table for Modbus slave response when the function code is 0x04, showing error codes and their meanings.

通信协议

- 007系列表使用Modbus RTU通信协议,进行RS485半双工通信。读功能号0x03,写功能号0x06,采用16位CRC校验,仪表对校验错误不返回。
通信异常处理:异常应答时,将功能号的最高位置1。例如:主机请求功能号0x04,则从机返回的功能号对应项为0x04。
错误应答码:
0x01—功能码非法:仪表不支持接收到的功能号。
0x02—数据位置非法:主机指定的数据位置超出仪表的范围。
0x03—数据值非法:主机发送的数据值超出仪表对应的数据范围。

Tables for Modbus communication: Host request (读多寄存器), Slave response (读多寄存器), and Slave response (写多寄存器).

功能号异常应答

Table for Modbus slave response when the function code is 0x04, showing error codes and their meanings.

写多寄存器

Table for Modbus slave response when writing multiple registers, showing error codes and their meanings.

从机正常应答

Table for Modbus slave response when the function code is 0x04, showing data format and error codes.

数据位置错误应答

Table for Modbus slave response when the function code is 0x04, showing error codes and their meanings.

通信协议

- 007系列表使用Modbus RTU通信协议,进行RS485半双工通信。读功能号0x03,写功能号0x06,采用16位CRC校验,仪表对校验错误不返回。
通信异常处理:异常应答时,将功能号的最高位置1。例如:主机请求功能号0x04,则从机返回的功能号对应项为0x04。
错误应答码:
0x01—功能码非法:仪表不支持接收到的功能号。
0x02—数据位置非法:主机指定的数据位置超出仪表的范围。
0x03—数据值非法:主机发送的数据值超出仪表对应的数据范围。

Tables for Modbus communication: Host request (读多寄存器), Slave response (读多寄存器), and Slave response (写多寄存器).

功能号异常应答

Table for Modbus slave response when the function code is 0x04, showing error codes and their meanings.

写多寄存器

Table for Modbus slave response when writing multiple registers, showing error codes and their meanings.

从机正常应答

Table for Modbus slave response when the function code is 0x04, showing data format and error codes.

数据位置错误应答

Table for Modbus slave response when the function code is 0x04, showing error codes and their meanings.

继上表

Table listing reserved expansion functions and parameters, including CT secondary current, communication address, and various rate and interval settings.

保留费率扩展

```
16位CRC校验码获取程序
unsigned int Get_CRC(uchar*Buf,uchar num)
{
    unsigned int i;
    unsigned int wCrc=0xFFFF;
    for(i=0;i<num;i++)
    {
        wCrc=(unsigned int)(pBuf[i]);
        for(j=0;j<8;j++)
        {
            if(wCrc & 1) { wCrc>>=1; wCrc=0xA001;}
            else wCrc>>=1;
        }
    }
    return wCrc;
}
```