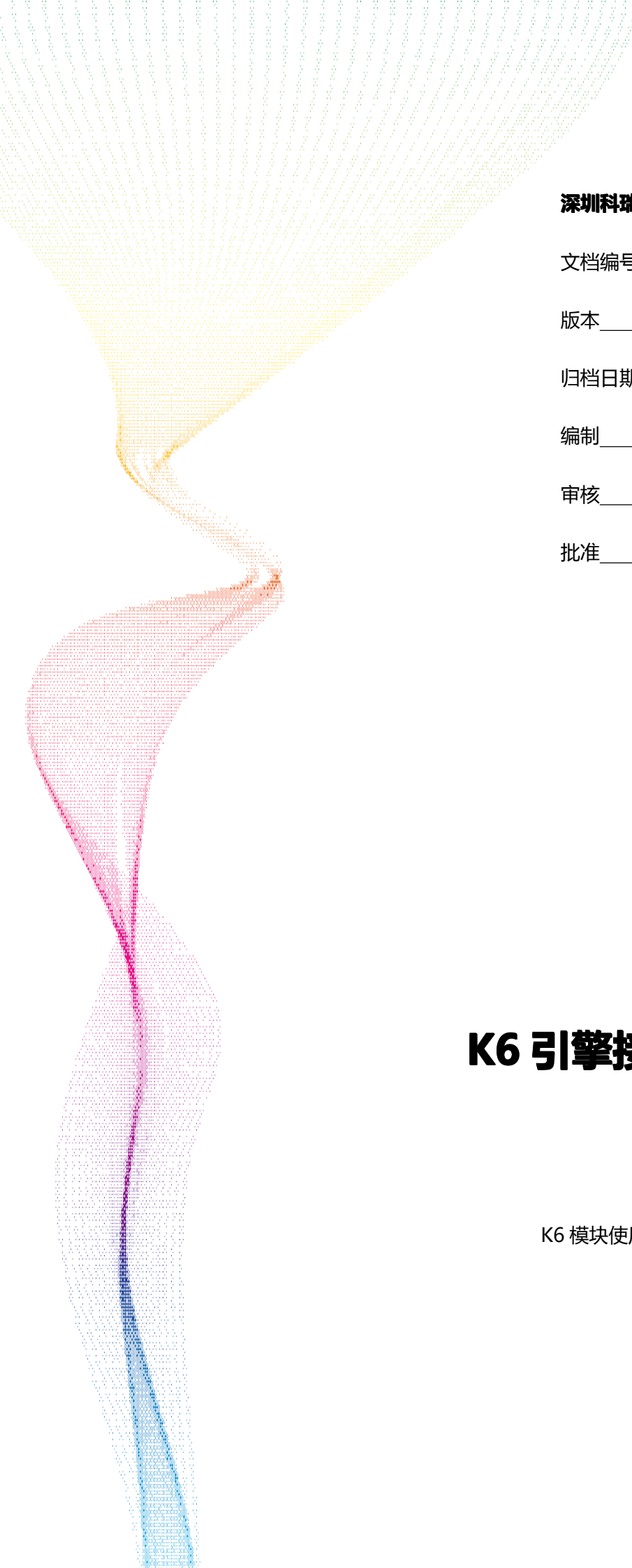




深圳科瑞格电子有限公司

文档编号 G5.07.K6.002.

版本 V2.7

归档日期 20120525

编制

审核

批准

K6 引擎接口设计说明书

Gump

K6 模块使用手册，供开发者阅读

[K6 引擎]接口设计说明书

0. 修改记录

V2.7

- 1、增加读取串号、写入串号的命令，写入串号之前须向 K6 发送进入校表状态命令，否则写入串号不会成功。
- 2、增加写入串号命令对应的动态链接库接口
- 3、修改了数字接口接收和发送引脚的错误
- 4、UART 的速度修改为 2400bps

1. 概述

本文档描述了 K6 引擎的对外硬件接口、软件接口和典型应用。

2. 指标

电压有效值：在 66V ~ 264V 之间， $\text{Err} \leq \pm(0.5\%+0.1\text{V})$

电流有效值：在 10mA ~ 120A 之间， $\text{Err} \leq \pm(0.5\%+1\text{mA})$

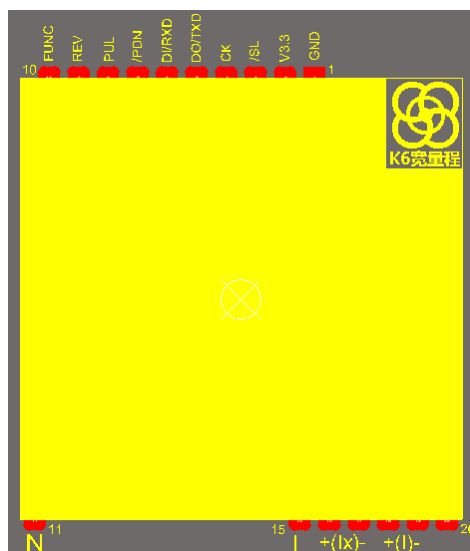
功率有效值： $\text{Err} \leq \pm(0.5\%+0.1\text{w})$

有功电能： $\text{Err} \leq \pm 0.5\%$

脉冲常数：3200imp/kWh

3. 硬件接口

K6 对外接口包含 2 个部分：数字接口和采样接口。



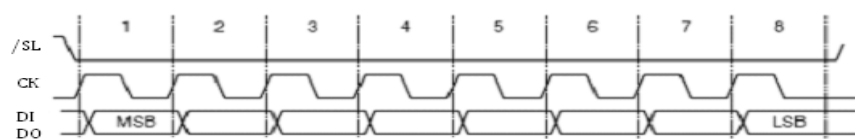
3.1. 数字接口

序号	标示	方向	含 义
1	GND	IN	电源地

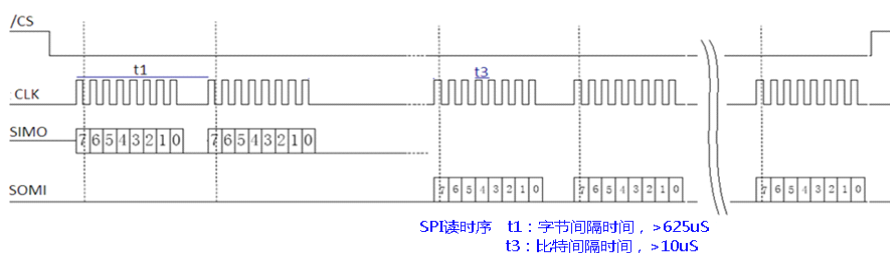
序号	标 示	方向	含 义
2	VCC	IN	电源, (3.0~3.3)V $\pm$ 5%
3	/SL	IN	SPI 接口选择使能, 低电平有效
4	CK	IN	SPI 接口时钟
5	DO/TXD	OUT	SPI 接口数据输出/UART 数据发送
6	DI/RXD	IN	SPI 接口数据输入/UART 数据接收
7	/PDN	--	未使用
8	PUL	OUT	有功脉冲高电平输出, 80ms 或 1:1 占空比, 最大脉冲速率 100Hz
9	REV ¹	OUT	有功方向, 0.5Hz 表示正向, 2Hz 表示反向
10	FUNC	IN	High 或悬空, 使用 SPI 接口; Low 使用 UART 接口

所有信号均为 3.0/3.3V CMOS 标准, 在使用时请注意。

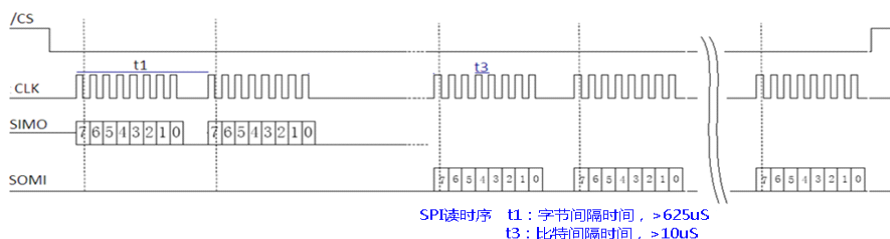
SPI 接口时序图:



SPI 读数据:



SPI 写数据:



UART 接口标准: 2400bps, 8 个数据位, 1 个偶效验位, 1 个起始位, 1 个停止位。

3.2. 采样接口

序号	标 示	方向	含 义
11	VN	IN	电压采样输入零线端
15	VL	IN	电压采样输入火线端
16	Ix+	IN	零线电流入(根据配置, 可能未使用)
17	Ix-	IN	零线电流出(根据配置, 可能未使用)
18	I+	IN	电流入
19	I-	IN	电流出
20	NC	--	未使用

1 未使用通信接口时, 可以用 REV 识别 K6 是否正常工作: 如果此信号电平保持不变, 则 K6 工作不正常。

4. 通信协议

可以通过 SPI/UART 访问 K6 引擎，对 K6 的访问有读当前数据、写参数、计量切换等命令几种操作。

4.1. 读当前数据

主机访问 K6 报文：**CMD**

CMD: 0x80 0xA5

K6 响应主机报文：**DATA(0) DATA(1) DATA(31) SUM8 XOR8**

DATA(0...31): 响应的当前数据块，各数据按低字节先传送，高字节后传送顺序

SUM8: 本报文除 SUM8 和 XOR8 外的各字节的算术和舍去 8 位字节以上的结果，下同

XOR8: 本报文除 SUM8 和 XOR8 外的各字节异或运算的结果，初值为 0x5A，下同

当前数据块定义：

u8 HWVer; //硬件版本

u8 SWVer; //软件版本

u8 bFlag; //bit0 校表模式, bit1 复位标志, bit2 有功电能方向, bit3 采用零线电流计量

u8 Pulse; //有功电能脉冲计数寄存器, 0~255 循环

u16 Urms; //电压有效值, 单位 0.01V

u32 Irms; //电流有效值, 单位 0.1mA

u32 IrmsAux; //零线电流有效值, 单位 0.1mA

s32 P; //有功功率, 单位 0.01W, 符号位表示方向

s32 PAux; //零线有功功率, 单位 0.01W, 符号位表示方向

u16 Freq; //电网频率, 单位 0.001Hz

s16 Ppf; //功率因数, 单位 0.01%

s16 PpfAux; //零线功率因数, 单位 0.01%

u16 PulseA; //正向有功电能脉冲计数寄存器, 0~65535 循环

u16 PulseR; //反向有功电能脉冲计数寄存器, 0~65535 循环

4.2. 读串号

主机访问 K6 报文：**CMD**

CMD: 0x87 0xA8

K6 响应主机报文：**DATA(0) DATA(1) DATA(11) SUM8 XOR8**

DATA(0...11): 串号

4.3. 写参数

K6 自动存储校表参数，不需要每次上电后重复执行写参数命令。

主机访问 K6 报文：**CMD DATA(0) ... DATA(N-1) SUM8 XOR8**

K6 响应主机报文：无

只有在 K6 处于校表模式下写参数命令才会被执行。

CMD	DATA	含 义
0x40	0x1357	初始化参数，校表参数回到缺省值
0x41	u16 Ugain	电压增益
0x48	u16 Igain[0]	小端电流增益
0x49	u16 Pgain[0]	小端电流功率增益

CMD	DATA	含 义
0x4a	s16 Phase[0]	小端电流角度校准
0x42	u16 lgain[1]	大端电流增益
0x43	u16 Pgain[1]	大端电流功率增益
0x44	s16 Phase[1]	大端电流角度校准
0x55	u16 lgainAuxL[0]	零线小端电流增益
0x56	u16 PgainAuxL[0]	零线小端功率增益
0x57	s16 PhaseAuxL[0]	零线小端角度校准
0x45	u16 lgainAux[1]	零线大端电流增益
0x46	u16 PgainAux[1]	零线大端功率增益
0x47	s16 PhaseAux[1]	零线大端角度校准
0x6e	0x8ace	输入小端电流后校正小端电流偏移
	0xace2	输入大端电流后校正大端电流偏移
0x6d	0x147a	零线输入小端电流后校正零线小端电流偏移
	0x47ad	零线输入大端电流后校正零线大端电流偏移
0x70	SN[12]	写入串号

4.4. 控制命令

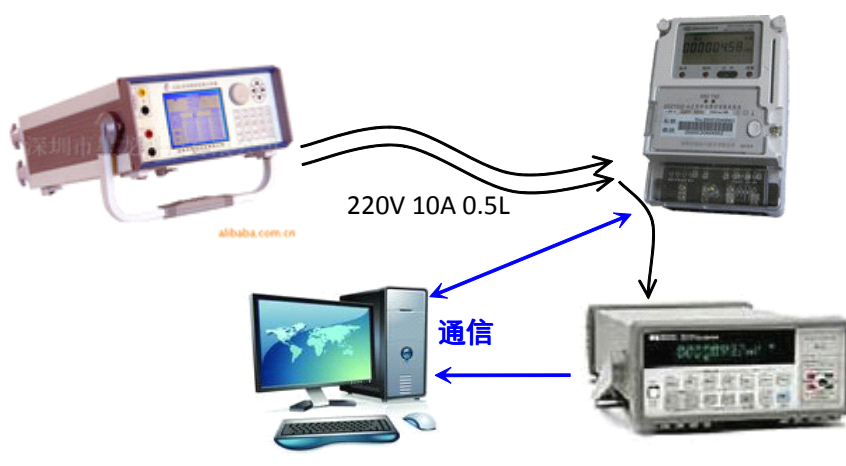
主机访问 K6 报文: **CMD DATA(0) DATA(1) SUM8 XOR8**

K6 响应主机报文: 无

CMD	DATA	含 义
0x6f	0x468a	进入校表模式, 重新上电后回到非校表模式
	≠ 0x468a	退出校表模式
0x5d	0x7654	使用零线回路计量
	≠ 0x7654	使用主回路计量

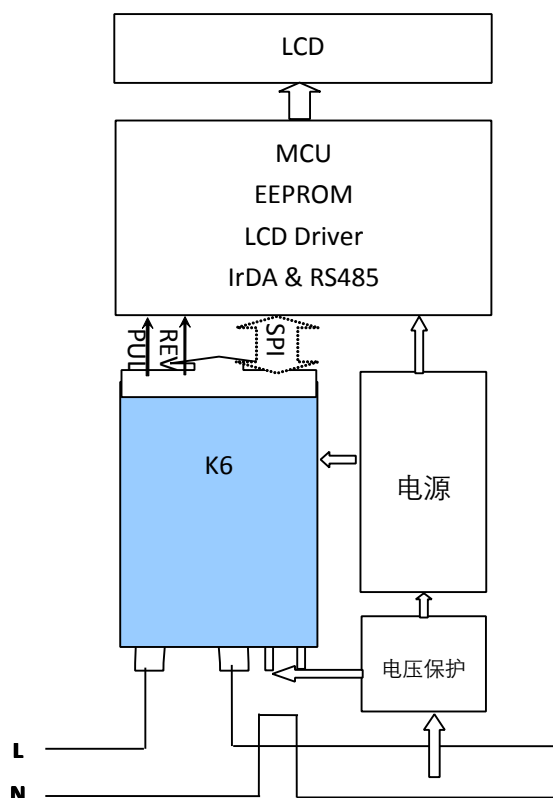
5. 校表实现

如下图, 标准源输入稳定的信号给被校电表, 同时也提供给标准表, 如果是表源一体化仪表, 则不需要标准表, 在计算机上运行校表软件 K6ADJ_NEW.EXE, 与被校电表通信。



K6ADJ_NEW.EXE使用方法参见该软件使用手册。

6. 典型应用



如果不需要瞬时量测量数据，则 SPI/UART 可以不接入。

K6 模块的位置铺地，并且加大面积丝印防止与 K6 上的线路短路，模块位置最好不要走线。

SPI 口、UART 口串 10 欧姆的电阻。

给 K6 供电处就近一个 47uF 以上的电容。

7. 校表 DLL 使用

校表需要发送相关的报文给 K6，发送的报文可以通过调用 K6ADJ_DLL.dll 的函数得到，下面是 DLL 函数接口的描述。

调用以下函数得到报文后，再将报文发送给 K6，即可实现对应的功能。

如果在发送者(如计算机)和 K6 之间的通信信道上有其他的 CPU，这时需要该 CPU 转发报文，转发方式通常有 2 种：**直接透明转发**和**打包转发**。

直接透明转发：计算机发送报文给 CPU，CPU 原文转发给 K6；CPU 收到 K6 报文，CPU 原文转发给计算机。

打包转发：计算机按照与 CPU 约定的格式将报文打包发送给 CPU，CPU 收到后去掉打包信息，将原始报文发送给 K6；K6 发送报文给 CPU，CPU 按照约定的格式打包发送给计算机。

```
//进入校表模式
```

//CmdW: 控制字, 0x468a 为进入校表模式, 否则进入非校表模式

```
//发送报文返回在 Packet[] 中
```

//返回报文长度

```
unsigned char K6ADJ_InFactory(unsigned short CmdW, unsigned char *Packet);
```

```
//初始化校表参数
```

```
//发送报文返回在 Packet[] 中
```

//返回报文长度

unsigned char K6ADJ_Init(unsigned char *Packet);

//计算电压通道增益

//UStd: 标准表的电压值, UK6: K6 测量的电压值, 单位: 0.01V

//发送报文返回在 Packet[]中

//返回报文长度

unsigned char K6ADJ_Ugain(unsigned long UStd, unsigned long UK6, unsigned char *Packet);

//设置小端电流通道偏置

//发送报文返回在 Packet[]中

//返回报文长度

unsigned char K6ADJ_loffset(unsigned char *Packet);

//IStd: 标准表的小端电流值, IK6: K6 测量的小端电流值, 单位: 0.1mA

//发送报文返回在 Packet[]中

//返回报文长度

unsigned char K6ADJ_lgain(unsigned long IStd, unsigned long IK6, unsigned char *Packet);

//计算小端电流有功功率增益

//Err: 小端电流有功电能误差, 单位: 0.001%

//发送报文返回在 Packet[]中

//返回报文长度

unsigned char K6ADJ_Pgain(signed long Err, unsigned char *Packet);

//计算小端电流角度校准

//Err: 小端电流有功电能误差, 单位: 0.001%

//发送报文返回在 Packet[]中

//返回报文长度

unsigned char K6ADJ_PhaseL(signed long Err, unsigned char *Packet);

//设置大端电流通道偏置

//发送报文返回在 Packet[]中

//返回报文长度

unsigned char K6ADJ_loffset(unsigned char *Packet);

//计算大端电流通道增益

//IStd: 标准表的电流值, IK6: K6 测量的电流值, 单位: 0.1mA

//发送报文返回在 Packet[]中

//返回报文长度

unsigned char K6ADJ_lgain(unsigned long IStd, unsigned long IK6, unsigned char *Packet);

//计算大端电流有功功率增益

//Err: 大端电流有功电能误差, 单位: 0.001%

//发送报文返回在 Packet[]中

//返回报文长度

unsigned char K6ADJ_Pgain(signed long Err, unsigned char *Packet);

//计算大端电流角度校准
//Err: 大端电流有功电能误差, 单位: 0.001%
//发送报文返回在 Packet[]中
//返回报文长度

unsigned char K6ADJ_Phase(signed long Err, unsigned char *Packet);

//设置零线小端电流通道偏置
//发送报文返回在 Packet[]中
//返回报文长度

unsigned char K6ADJ_loffsetAuxL(unsigned char *Packet);

//计算零线小端电流通道增益
//IStdAux: 标准表的电流值, IK6Aux: K6 测量的零线电流值, 单位: 0.1mA
//发送报文返回在 Packet[]中
//返回报文长度

unsigned char K6ADJ_lgainAuxL(unsigned long IStdAux, unsigned long IK6Aux, unsigned char *Packet);

//计算零线小端电流有功功率增益
//ErrAux: 零线小端电流有功电能误差, 单位: 0.001%
//发送报文返回在 Packet[]中
//返回报文长度

unsigned char K6ADJ_PgainAuxL(signed long ErrAux, unsigned char *Packet);

//计算零线小端电流角度校准
//ErrAux: 零线小端电流有功电能误差, 单位: 0.001%
//发送报文返回在 Packet[]中
//返回报文长度

unsigned char K6ADJ_PhaseAuxL(signed long ErrAux, unsigned char *Packet);

//设置零线大端电流通道偏置
//发送报文返回在 Packet[]中
//返回报文长度

unsigned char K6ADJ_loffsetAux(unsigned char *Packet);

//计算零线大端电流通道增益
//IStdAux: 标准表的电流值, IK6Aux: K6 测量的零线电流值, 单位: 0.1mA
//发送报文返回在 Packet[]中
//返回报文长度

unsigned char K6ADJ_lgainAux(unsigned long IStdAux, unsigned long IK6Aux, unsigned char *Packet);

//计算零线大端电流有功功率增益
//ErrAux: 零线大端电流有功电能误差, 单位: 0.001%
//发送报文返回在 Packet[]中
//返回报文长度

unsigned char K6ADJ_PgainAux(signed long ErrAux, unsigned char *Packet);

//计算零线大端电流角度校准

//ErrAux: 零线大端电流有功电能误差, 单位: 0.001%

//发送报文返回在 Packet[]中

//返回报文长度

unsigned char K6ADJ_PhaseAux(signed long ErrAux, unsigned char *Packet);

//计量切换

//CmdW: 控制字, 0x7654 为使用零线计量, 否则使用主回路计量

//发送报文返回在 Packet[]中

//返回报文长度

unsigned char K6ADJ_PhaseAux(unsigned short CmdW, unsigned char *Packet);

//写入串号

//SN[12]: 串号

//发送报文返回在 Packet[]中

//返回报文长度

unsigned char K6ADJ_WriteSN(unsigned char *SN, unsigned char *Packet);