手持POS开发手册

Ver 4.2

目　录

一、前言

DH 系列手持机具备高性能 32 位 CPU,大容量 FLASH 存储,大屏幕显示。提供极易开发的 C 语言二次开发平台，丰富接口函数，用户可按自己需求编写应用程序。采用先进的 FLASH 管理算法，支持文件系统（8.3 文件名）和 DBF 数据库操作，数据存储安全可靠，和 PC 交换数据更简单方便。支持多种通讯方式（RS232，USB，IRDA，无线），可选非接触卡读头，支持 ISO14443A/B、ISO15693 卡片读写。

DH 系列手持机广泛应用于水、电、气抄表，仪器仪表数据采集，小区收费管理，身份验证，移动售饭、考勤、门禁,巡更/巡检,咪表管理,会员制连锁消费,烟草专卖、稽查，税务系统电子报税，儿童疫苗接种管理，商场盘点等。

本手册说明手持机提供的接口函数以及WINDOWS平台下动态库调用接口函数，用户可以利用接口函数编写符合自己要求的应用程序,在编写程序前仔细阅读本册。

本手册提供的函数接口适用于我公司32位平台开发的各种产品，用户无须重新开发即在其他产品运行，保证用户的开发资源合理利用。

二、开发要求

硬件要求：IBM-PC机586-133，16MB内存，CDROM。
软件要求：WINDOWS操作系统。

适用POS：DH2000系列，DH3000系列

三、主要技术参数

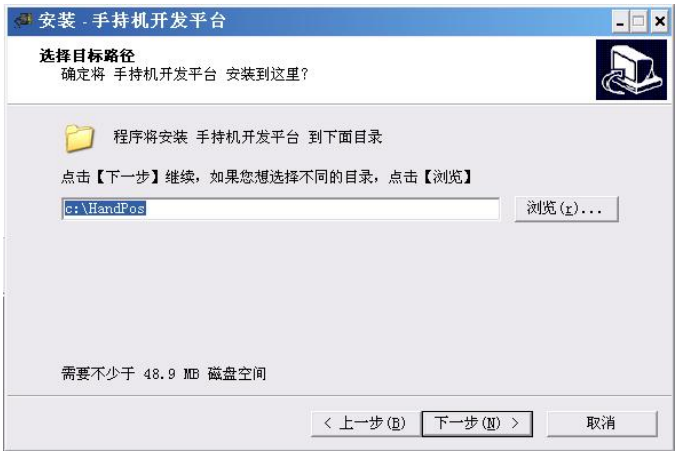
DH系列手持机采用相同的硬件核心，用户的应用程序可以在不同型号的手持机上运行，满足用户不同需求，主要的硬件参数如下（不同型号的手持机参数会有差异）

C P U	32 位 ARM CPU
内　存	2M Byte FLASH（可选 8M Byte FLASH） 512K Byte SRAM,
显　示	160*160 全点阵液晶，EL 背光，一屏可显示 20*20 字符或 10*10 汉字
字　库	国标二级汉字库
按　键	20 按键，支持拼音输入
非接触卡	支持 ISO14443A/B，ISO15693（可读写卡片：mifre,i.code,ti.tag）
通　讯	标准 RS－232 接口(可选 USB 接口、IRDA 红外线通讯口)
二次开发	C 语言开发系统，支持文件系统和 DBF 数据库操作
电　源	锂电池充电电池
功　耗	工作时 15mA，待机时小于 300uA
温　度	-5℃－60℃
相对温度	45%－95%
抗磁场干扰	19 奥思特
抗静电干扰	5KV

四、安装开发环境

1、双击 setup.exe 开始安装编译器，执行后，如下图：

2、选择编译器安装目录，如下图。建议使用默认安装目录，在后面的集成开发环境中均使用默认的开发目录



3．选择程序菜单的目录

4．创建桌面快捷方式


安装 - 手持机开发平台
选择附加任务
哪一个附加任务将被执行?
选择当安装 手持机开发平台 完成时，您希望被执行的附加任务，然后点【下一步】
附加图标:
创建桌面图标
< 上一步(B)
下一步(N) >
取消

6、开始安装


安装 - 手持机开发平台
安装
请等待，程序正在安装 手持机开发平台 到你的计算机上
解压文件...
c:\HandPos\complier\lib\gcc-lib\arm-elf\3.3.1\cc1plus.exe
取消

7、安装成功


安装 - 手持机开发平台
手持机开发平台 安装向导完成
安装程序已经将 手持机开发平台 安装到您的计算机上。您可以通过点击相应的图标以启动安装好的程序
点击【完成】退出安装
完成(F)

五、编译环境设置

手持机二次开发平台采用流行的GNU编译器作为开发工具，当前版本3.3.1.使用WIN IDE作为开发核心。具有代码编辑，工程管理，C语法高亮提示，函数自动提示，手持机程序下载和管理等特色功能，使手持机开发更容易。

1、 重新设置编译器目录

编译器在安装时已经将默认参数设置好，无须设置编译环境。如更改编译器的安装目录，必须重新设置编译。选择IDE菜单 工具->编译器选项->目录->二进制，界面如下：

选择删除后，重新定位新的目录，点击添加即可

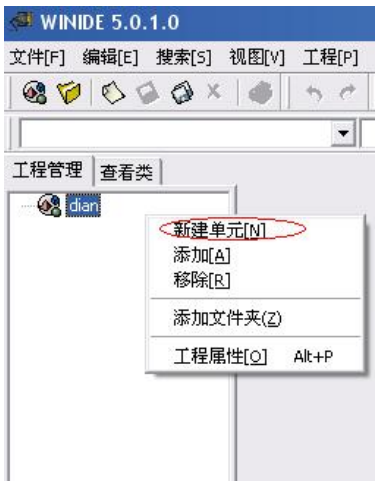
2、 新建工程

选择IDE菜单 文件->新建->工程 ,界面如下:

键入工程名和选择默认工程模板，点确定，将工程保存在适当的位置，工程新建后，编译器参数以设置为手持机开发模式，无须做其他更多设置。

3、 添加项目文件

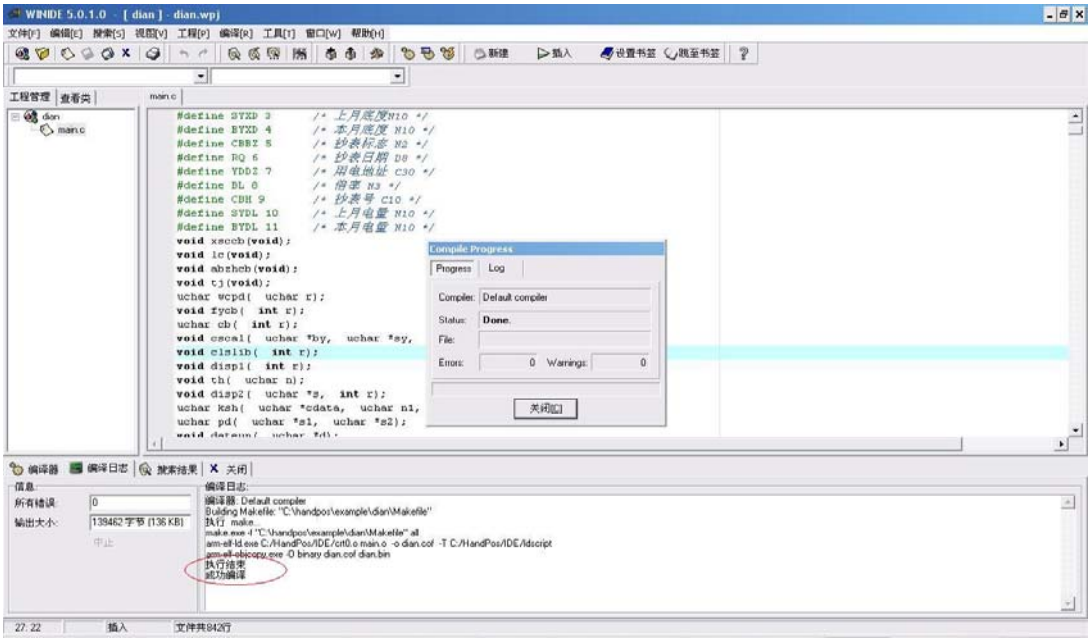
右击工程管理中的工程名，如图:



选择新建单元或添加已存在的文件。

4、 编译当前工程

选择IDE菜单 编译->编译，编译提示如下:



5、 设置手持机通讯端口：

选择IDE菜单 工具->环境选项->基本,如图：

6、 编译下载

手持机可执行的文件为.bin的文件，编译器编译成功后，生成手持机可执行的文件。选择 编译->下载，即可下载文件，若连

接不成功，检查通讯端口是否选择正确，手持机是否进入通讯状态、通讯线是否连接正确。

六、开发函数库说明

在使用这些库函数时，请先包含 Hspos.h头文件。

6.1 键盘函数

6.1.1 unsigned char key(unsigned char k)

描述: 从键盘等待接收一个按键,.
入口: k =0,按键无回显;
k=1,按键带回显(在当前光标位置显示)
返回: 键值

例子: #include “Hspos.h”
xdata k;
k=key(0);

6.1.2 unsigned char keys(unsigned char *s)

描述: 从键盘接收一串字符,带回显.
入口: s键值缓冲区指针
返回: 字符串长度
说明: 在按键输入字符串的过程中,如果输错可以按[清除]键删除上一输入,按[确认]键后退出接收.且自动在串s的末尾加结束符0x00

6.1.3 unsigned char keysn(unsigned char *s,unsigned char n)

描述: 从键盘接收一串字符,带回显.
入口: s 键值缓冲区指针
n 接收的有效字符长度
返回: 字符串长度

6.1.4 unsigned char keygo(void)

描述: 从键盘接收一个按键,不等待.
入口: 无
返回: 键值
说明: 此函数一般用于查询是否有键按下，而无须等待。

拼音输入使用说明:

手持机采用类T9拼音输入方法，使用方法如下：

1. 需要在程序中调用KEYS或KEYSN，接受拼音、字母大写、字母小写输入
例如
unsigned char kdata[100],l;
l=keys(kdata);

2. 按[F1]切换输入法，在屏幕右下脚提示当前输入状态，默认为数字输入.
状态提示：
123，数字 ；
abc 字母小写（包含数字输入）；
ABC 字母大写（包含数字输入）；
PY，拼音输入

3. 按[清除]拼音输入，确认结束输入，
   按‘.’选择拼音，‘+’选择当前汉字，
   上翻，下翻，选择汉字行选择
   左翻，右翻，选择汉字的位置

4. 拼音输入状态
   采用T9拼音联想输入，只需要输入字母所对应的数字，如输入拼音 ming，
   则只需要按4次数字键 6464

6.2 显示接口函数

6.2.1 void screen(unsigned char bz)

描述: 设置显示状态
入口: bz 0--字符状态
1--汉字状态
返回: 无
说明: 在用函数putn(),putstr()等函数显示字符串时,如果字符串中有汉字,则应该在调用之前将显示状态设为汉字状态,即:screen(1);

6.2.2 void cls(void)

描述: 清屏
入口: 无
返回: 无

6.2.3 void clsn(unsigned char lin, unsigned char n)

描述: 清除屏幕上从lin行开始的n行显示.
入口 : lin 要清除的起始行号(1--20)
n 清除的行数(1--20)
返回: 无

6.2.4. void moveto(unsigned char x, unsigned char y)

描述: 移动光标

入口:　　x　　行(1-20)
　　y　　列(1--20)

6.2.5　unsigned　char getx(void)

描述:　　返回当前光标的行号(1--20)

6.2.6　unsigned　char gety(void);

描述:　　返回当前光标的列号(1-20)

6.2.7　void putch(unsigned　char c)

描述:　　显示一个字符.
入口:　　c　　字符ASCII码

6.2.8　void putstr(unsigned　char *s)

描述:　　显示一串字符或汉字.
入口　　:s　　字符串指针
说明:　　如果串中有汉字,应先将显示状态置为'汉字状态'(screen(1)),
　　参数s可以串中可以带'\n'参数 作为换行符.串应以\0结束.
例子:

```
①putstr("汉字显示");
②unsigned　char s[8];
  gettime(s);
  putstr(s);
```

6.2.9　void putn(unsigned　char n, unsigned　char *s)

描述:　　显示n个字符或汉字.
入口:　　n　　字符数
　　s　　字符串指针
说明:　　如果串中有汉字,应先将显示状态置为'汉字状态'(screen(1)),
　　一个汉字算两个字符.
例子:

```
unsigned　char s[8];
putstr("当前时间:");
```

```
gettime(s);
putn(8,s);
```

6.2.10 void putch_h(unsigned char c)

描述: 显示一个字符.
入口: c 字符ASCII码
说明: 该函数与putch()的唯一区别是显示的字符是全高字符.

6.2.11 void putstr_h(unsigned char *s)

描述: 显示一串汉字或字符.
入口: s 字符串指针
说明: 该函数与putstr()的唯一区别是显示的字符是全高字符.

6.2.12 void putn_h(unsigned char, unsigned char *)

描述: 显示n个字符或汉字.
说明: 该函数与putn()的唯一区别是显示的字符是全高字符.

6.2.13 void setdot(unsigned char x unsigned char y)

描述: 点亮屏幕上某一点.
入口: x 行坐标(1--160)
y 列坐标(1--160)

6.2.14 void clrdot(unsigned char x unsigned char y)

描述: 清除屏幕上某一点.
入口: x 行坐标(1--160)
y 列坐标(1--160)

6.2.15 void put_block(U8 x,U8 y,U8 *s);

描述: 写显示缓冲8x8的自定义数据
入口: x 行坐标(1--160)
y 列坐标(1--160)

s　　8x8 个字节

6.2.16 void get_block(U8 x,U8 y,U8 *s);

描述:　　得到显示缓冲8x8的数据
入口:　　x　　行坐标(1--20)
　　y　　列坐标(1--20)
　　s　　8x8 个字节

6.2.17 void reverse_block(U8 x,U8 y);

描述:　　反显显示缓冲上8x8的数据
入口:　　x　　行坐标(1--20)
　　y　　列坐标(1--20)
　　s　　8x8 个字节

6.2.18 void save_video(U8 sx,U8 sy,U8 ex,U8 ey,U8 *buf_ptr);

描述:　　保存显示缓冲上某一块的数据
入口:　　sx,sy　　块开始坐标(1--160)
　　ex,ey　　块结束坐标(1--160)

6.2.19 void restore_video(U8 sx,U8sy,U8 ex,U8 ey,U8 *buf_ptr);

描述:　　回写显示缓冲上某一块的数据
入口:　　sx,sy　　块开始坐标(1--160)
　　ex,ey　　块结束坐标(1--160)

6.2.20 void drawline(U8 sx,U8 sy,U8 ex,U8 ey);

描述:　　画线，不支持斜线
入口:　　sx,sy　　线开始坐标(1--160)
　　ex,ey　　线结束坐标(1--160)

6.2.21 void drawrect(U8 sx,U8 sy,U8 ex,U8 ey);

描述:　　画矩形
入口:　　sx,sy　　矩形始坐标(1--160)

ex,ey　　线结束坐标(1--160)

sx,sy

ex,ey

6.2.22 void putch_x(　U8 dischar,U8 dis_mode);

描述:　　显示一个字符.
入口:　　dischar　　字符 ASCII 码
　　dis_mode　　显示模式　=0，正显；=1，反显

*6.2.23 void putstr_x(　U8 *disstr,U8 dis_mode);*

描述:　　显示一串字符或汉字.
入口　　*disstr　　字符串指针
　　dis_mode　　显示模式　=0，正显；=1，反显
说明:　　如果串中有汉字,应先将显示状态置为'汉字状态'(screen(1)),
参数*disstr 可以串中可以带'\n'参数 作为换行符.串应以\0 结束.

*6.2.24 void putn_x (U8 strlen,　U8 *disstr,U8 dis_mode);*

描述:　　显示一串字符或汉字.
入口　　strlen　　显示长度
　　*disstr　　字符串指针
　　dis_mode　　显示模式　=0，正显；=1，反显
说明:　　如果串中有汉字,应先将显示状态置为'汉字状态'(screen(1)),
参数 s 可以串中可以带'\n'参数 作为换行符.串应以\0 结束.

6.3 时钟接口函数

6.3.1 void gettime(unsigned char *t)

描述: 取时间
入口: *t hh.mm.ss(时.分.秒)形式的字符串,8个字符.

6.3.2 void getdate(unsigned char *d)

描述: 取日期
入口: *d yyyy.mm.dd(年-月-日)形式的字符串,10个字符.

6.3.3 unsigned char getweek(void)

描述: 取星期
入口: 无
返回: 星期几.

6.3.4 void settime(unsigned char *t)

描述: 设置时间
入口: *t hh.mm.ss(时.分.秒)形式的字符串,8个字符.
例子:

```
设置时间为12时5分5秒
settime("12-05-05");
或: unsigned char s[10];
settime(s);
```

6.3.5 void setdate(unsigned char *d)

描述: 设置日期
入口: *d yyyy.mm.dd(年-月-日)形式的字符串,10个字符.
出口: 无

6.4 文件系统

POS机文件系统（FLASH filesystem）采用先进的FLASH管理算法来均衡FLASH擦写次数，可以管理128个文件。使用户不需要关心FLASH底层操作，通过通讯管理工具和PC交换数据。文件名采用8.3的格式，超过此长度时POS机将自动截取。

POS机可以执行的文件后缀为.bin，其他文件流与PC机相同

6.4.1 int FS_creatfile(char *filename,char attr);

描述:	建立一个文件	
入口:	*filename	文件名，采用8.3的格式
	attr	文件属性 0
返回:	0	正确
	其他值	错误

说明: 在文件不存在时先用此文件建立一个空文件

6.4.2 int FS_deletefile(char *filename);

描述:	删除文件	
入口:	*filename	文件名
返回:	0	正确
	其他值	错误

6.4.3 int FS_find(char *filename);

描述:	查找文件是否存在	
入口:	*filename	文件名
返回:	0	正确
	其他值	错误

说明: 无

6.4.4 int FS_geterror(void);

描述:	返回操作文件错误时的系统错误值
入口:	无
返回:	错误值

6.4.5 FS_FILE *FS_fopen(char * /*filename*/, char * /*mode*/);

描述: 打开一个存在在的文件
入口: *filename 文件名
mode 打开文件模式
'r' 只读
'w' 写
'a' 追加

返回: 0 正确
其他值 错误

6.4.6 int FS_fclose(FS_FILE * /*stream*/);

描述: 关闭已经打开的文件
入口: * stream 文件句柄
返回: 0 正确
其他值 错误

6.4.7 size_t FS_fread(void *,size_t , size_t , FS_FILE *);

描述: 读文件
入口: *ptr 数据缓冲区
size 读长度
nmemb 读块大小
stream 文件句柄
返回: 0 正确
其他值 错误

6.4.8 size_t FS_fwrite(void * ,size_t, size_t , FS_FILE *);

描述: 写文件
入口: *ptr 数据缓冲区
size 写长度
nmemb 写块大小
stream 文件句柄
返回: 写入长度 写入成功
0 写入错误

6.4.9 int FS_fseek(FS_FILE * /*stream*/, long int /*offset*/, int /*whence*/);

描述: 文件指针定位
入口: * stream 文件句柄
offset 指针偏移长度
whence 相对位置
返回: 0 正确
其他值 错误
说明:
whence 位置值
#define FS_SEEK_SET 0 /* set file offset to offset */
#define FS_SEEK_CUR 1 /* set file offset to current plus offset */
#define FS_SEEK_END 2 /* set file offset to EOF plus offset */

6.4.10 int FS_feof(FS_FILE * /*stream*/);

描述: 判断文件是否到文件尾
入口: * stream 文件句柄
返回: 0 是在文件尾
1 不在文件尾
其他值 错误操作

6.5 DBF 数据库操作

POS机提供一整套操作DBF数据库的函数，支持FOXPROIII,DBASE III以下标准的DBF数据库，PC通过通讯软件下载DBF文件到POS机后，POS可以根据DBF文件名来实现改写，追加，删除记录，最多可以同时打开16个DBF数据库

操作数据库字段时，利用字段序列号来定位字段。定位记录和字段的函数为libset，详见其说明

6.5.0 unsigned char libuse(unsigned char *filename,unsigned char n)

*描述:*打开filename指定的dbf数据库，打开顺序为n
入口: *filename,dbf数据库名
n 数据库序号(0-15 :表示第一至第十六数据库，可最多打开16个数据库)
*返回:*0 成功
其他 失败

*说明:*此函数目的是与H8000保持兼容而设的，和libopen配合使用

```
 rt=libuse("dian.dbf",0);       //打开dian.dbf数据库，顺序为一
if(rt!=0)
```

```
{
 putstr("打开数据库错误");
 bell(50);
 key(0);
}
libopen(0);//设置当前操作数据库，序号为一，与libuse()第二个参数保持一致
```

6.5.1 unsigned char libopen(unsigned char n)

*描述:*打开第n个数据库作为当前库.
*入口:*n 数据库号(0-15 :表示第一至第十六数据库)
*返回:*0 成功
1 无此库
2 库号超出0-15
*说明:*如果数据库不能打开,系统会报错.

6.5.2.unsigned int libsumr(void)

*描述:*返回当前数据库的记录总数.

6.5.3.unsigned char libsumf(void)

*描述:*返回当前数据库的字段总数.

6.5.4 unsigned int libgetr(void)

*描述:*返回当前的记录序号.
*说明:*记录序号从0开始,表示第一个记录,下同.

6.5.5.unsigned char libgetf(void)

*描述:*返回当前的字段序号.
*说明:*字段序号从0开始,表示第一个字段,下同.

6.5.6.unsigned char libset(int n, unsigned char n)

描述:设置当前记录序号和字段序号.
入口:n 记录序号
m 字段序号
返回:0 序号设置正确
1 序号超出范围
说明:如果序号超出范围,则设置为当前库的第一记录的第一字段.

例：Libset(100,8);定位在第100条记录的第9个字段（以0开始），在操作记录前都要利用此确定当前的记录和字段是否为所需的位置

6.5.7.unsigned char libread(unsigned char *s)

描述:读当前字段.
入口:s 存放读出内容的缓冲区指针
返回:读出串的长度
说明:读出的数据串已将字段中实际数据头尾空格删除,故返回的长度值小于或等于当前字段的字段长.如果字段为空,则返回0x20

6.5.8 void libwrite(unsigned char *s)

描述:写当前字段.
入口:s 存放欲写内容的缓冲区指针
说明:该函数将根据数据库中当前字段的类型及长度对写入数据作相应调整.

6.5.9 void libdel(void)

描述:删除当前记录.
说明:运行该函数前要用libset()函数设置当前记录.

6.5.10 void libapp(void)

描述:在数据库尾追加一条空记录.

6.5.11 int libreadrec(unsigned char *s);

描述:读当前记录.
入口:s存放读出内容的缓冲区指针，采用二维指针，第一维为字段号，第二维字段内容
返回:读出串的长度
说明:一次读出一整条记录
具体参考读写 DBF 数据库的例子

6.5.12 int libwriterec(unsigned char *s);

描述:写当前记录.
入口:s 存放欲写内容的缓冲区指针，采用二维指针，第一维为字段号，第二维字段内容
说明:一次写入一条记录 ，一个字段分配32个字节（31个字节有效信息+结束符号=32字节）
具体参考读写 DBF 数据库的例子

6. 7 字符串计算

支持字符串直接加、减、乘、除，如 “123.11”与”7.8”可直接运算,配合DBF数据库字段计算操作,参与运算的字符串必须带结束符(0x00)。

6.7.1.unsigned char stradd(uchar *d, uchar *s, uchar n)

描述:两个数字字符串相加(减).
入口:d 第一个串指针
s 第二个串指针,计算结果也保存此处.
n 字符串长度
返回:0 计算成功
1 溢出
说明:字符串可以带有'-'号及小数点.字符串的格式要求左对齐,长度不及n的在余下位置赋空字符(0x20).计算的结果程序会自动按此格式办理.

例子：
```
unsigned char s1[20];
unsigned char s2[20];
strcpy(s1,”123.11”);
strcpy(s2,”7.8”);
stradd(s1,s2,15);   函数执行后S2将返回”130.91”
```

6.7.2 unsigned char strsub(uchar *d, uchar *s, uchar n)

描述:两个数字字符串相减.
入口:d 第一个串指针
s 第二个串指针,计算结果也保存此处.
n 字符串长度
返回:0 计算成功
1 溢出
说明:字符串必须是无符号的.字符串格式同上.

6.7.3.unsigned char strmul(uchar *d,uchar *s,uchar n)

描述:两个数字字符串相加乘.
入口:d 第一个串指针
s 第二个串指针,计算结果也保存此处.
返回:0 计算成功
1 溢出
说明:字符串可以带有'-'号及小数点.字符串格式同上.

6.7.4.unsigned char strdiv(uchar *d, uchar *s,uchar n)

描述:两个数字字符串相加除.
入口:d 第一个串指针
s 第二个串指针,计算结果也保存此处.
返回:0 计算成功
1 溢出
说明:字符串可以带有'-'号及小数点.字符串格式同上.

6.8 通讯函数

6.8.1 void cominit(unsigned char baud,unsigned char ctrl,unsigned char ust);

描述:初始化通讯端口
入口:band 波特率,使用如下宏定义作为参数，说明如下:

```
B115200     /* 115200波特率  */
B57600      /* 57600波特率  */
B28800      /*28800波特率  */
```

B19200 /*19200波特率 */
B14400 /* 14400波特率 */
B9600 /* 9600波特率 */
B4800 /* 4800波特率 */
B2400 /* 2400波特率 */
B1200 /* 1200波特率 */

Ctrl 设置停止位，数据位和校验位
Bit1,Bit0: 数据位长度
00 5位
01 6位
10 7位
11 8位
Bit2 停止位长度
0 一位停止位
1 两位停止位
Bit5,4,3 奇偶校验
000 无校验
100 奇校验
101 偶校验
110 校验位强制为1
111 校验位强制为0
Bit6，7 00

ust ：打开或关闭通讯口
1 打开串口
0 关闭串口

说明:在各种方式下,数据都包括1位起始位,8位数据位,1位停止位.
例如：cominit(B115200 0x03, UART_ON);将串口设置为115200波特率，无校验

6.8.2.unsigned char comread(unsigned char *c)

描述: 从通讯口读一字符
入口:*c 接收的数据
返回:0 接收到一个数据
1 奇偶校验错

6.8.3.void comwrite (unsigned char c)

描述:向通讯口写一字符
入口:c 发送的数据

6.8.4. unsigned char combuf(void)

描述:读串口缓冲区中的数据
入口: 无
返回: 串口数据

6.8.5. unsigned char comstate(void)

描述:读串口状态
入口:无
返回 =0 无数据
1 有数据
说明 ：comstate 一般与combuf结合起来用实现非等待接收数据
例如：

```
while(1)
{
    if(comstate()!=0)
    {
        rbuf[0]=combuf();
        putchhex(rbuf[0]);
    }
    else break;
}
```

6.8.6 void ir_init (unsigned char baud,unsigned char ctrl,unsigned char ust);

描述: 初始化红外端口
入口:band 波特率,使用如下宏定义作为参数，说明如下:

```
B115200     /* 115200波特率 */
B57600      /* 57600波特率 */
B28800      /*28800波特率 */
B19200      /*19200波特率 */
B14400      /* 14400波特率 */
B9600       /* 9600波特率 */
B4800       /* 4800波特率 */
B2400       /* 2400波特率 */
B1200       /* 1200波特率 */
```

Ctrl 设置停止位，数据位和校验位
Bit1,Bit0: 数据位长度
00 5位
01 6位
10 7位
11 8位

Bit2　　停止位长度
　　2　一位停止位
　　3　两位停止位
Bit5,4,3　奇偶校验
　　000　　无校验
　　102　　奇校验
　　103　　偶校验
　　112　　校验位强制为1
　　113　　校验位强制为0
Bit6，7　　00

ust ：打开或关闭通讯口
　　1　　打开串口
　　0　　关闭串口

说明:在各种方式下,数据都包括1位起始位,8位数据位,1位停止位.
例如：ir_init (B115200 0x03, UART_ON);将红外口设置为115200波特率，无校验

6.8.7.unsigned　char ir_read (unsigned　char *c)

描述:从红外口读一字符
入口:*c　　接收的数据
返回:0　　接收到一个数据
　　1　　奇偶校验错

6.8.8.void ir_write (unsigned　char c)

描述:向红外口写一字符
入口:c　　发送的数据

6.8.9. unsigned char ir_rxbuf (void)

描述:读红外口缓冲区中的数据
入口: 无
返回: 红外口数据

6.8.10. unsigned char ir_rxstate (void)

描述:读红外口状态

入口:无
返回 =0　　无数据
　　1　　有数据
说明：ir_rxstate一般与ir_rxbuf结合起来用实现非等待接收数据
例如：

```
while(1)
{
    if(ir_rxstate ()!=0)
    {
        rbuf[0]= ir_rxbuf ();
        putchhex(rbuf[0]);
    }

}
```

6.9 打印函数

此部分函数适合我公司带打印系列机型，打印缓冲目前为十行汉字字符，当数据超过缓冲区时，超过的部分自动舍弃。字符点阵大小为24*24点阵，每行最多可以打印16个汉字(384点)

6.9.1 void Printer_SpaceSet(unsigned int spcx,unsigned char spcy);

描述:行列间距设定
入口:spcx(240)字符汉字间距设定，以点数表示，不包含字符宽度
　　spcy(0~100)行间距设定，以点数表示，含字符高度。
　　0<=spcx<240,24<=spcy<100
出口：无

6.9.2 unsigned char Printer_Detect(void);

描述:检测打印机状态
入口:无
出口：位为1指示有错,B3-B7保留

B7	B6	B5	B4	B3	B2	B1	B0
-	-	-	-	电压低	缓冲溢出	超温	无纸

最低位 表无纸，
第二位 超温,
第三位,缓冲溢出指示
第四位 电池电压不够
示例: 0x00　打印机正常
　　0x01　无纸

6.9.3 void Printer_PaperFeed(unsigned long line);

描述：进纸
入口：line:　　进纸的点行数
出口：无

6.9.4 void Printer_PaperBack(unsigned long line);

描述：退纸
入口：line:　　退纸的点行数
出口：无

6.9.5 void Printer_LF(unsigned int const lin);

描述：缓冲区换行
入口：换行间距(不包括字体所占的24点行)
出口：无

6.9.6 void Printer_Str(char const * ptr);

描述：打印字符串到缓冲区
入口：ptr　汉字字符串指针
出口：无
示例：　　Printer__Str("中华人民共和国12345678\n");
说明：　　字串可包含汉字及ASCII字符,不可显字符除0xa表示按设置值换行外均自动舍弃.
　　当当前行缓冲不足存放一个字符或汉字时自动换行。

6.9.7 void Printer_ClrBuf(void);

描述：清空打印缓冲区
入口：
出口：无
说明：清缓冲区后清除了缓冲溢出标志。

6.9.8 unsigned char Printer_Buffer(unsigned long const line);

描述：打印缓冲数据
入口：打印完成后继续进纸点行数
出口：0成功，非0时与Printer_Detect相同

6.9.9 unsigned char Printer_Open(void);

描述：打印机初始化函数
入口：无
出口：与Printer_Detect相同
说明：　　开机必须调用该函数进行打印初始化，至少一次。
　　包含了端口输入输出初始化及打印参数复位操作.

6.9.10 unsigned char Printer_Close(void);

描述:打印电源函数
入口: 无
出口:执行前的电源状态

6. 10 其他函数

6.10.1.unsigned　char itos(int i,unsigned　char *s)

描述:将整型数转换为ASCII码串.
入口:i　　整型数
　　*s　　转换后的ASCII串
返回:ASCII串的长度.

6.10.2.int stoi(unsigned　char n, unsigned　char *s)

描述:将ASCII码数字串转换为整数.
入口:n　　串长度
　　*s　　串
返回:转换后的整数.

6.10.3.unsigned char ltoa(long l, unsigned char *s)

描述:将长整数转换为ASCII码数字串.
入口:l 长整数
*s 转换后的ASCII数字串
返回:串长度

6.10.4.void bell(unsigned char t)

描述: 蜂鸣器响
入口:t 时长(1--200)

6.10.5 void Battery_display(U8 x,U8 y)

功能：电池电量显示
入口: x,y为显示坐标
出口：无

6.10.6 void des_code(U8 k, U8 *pdata, U8 * dkey, U8 *result);

功能：DES加密算法
入口：k=0为加密。/k=1为解密
*pdata 输入明文数据
* dkey 密钥
*result 输出密文数据
出口：无

6.10.7 U16 GET_Chip_ID(U8 *rid)

功能：得到POS机设备ID号，每台POS机都有个全球唯一的ID号，共16个字节
入口：无
出口：U8 *rid 16个字节的ID号缓冲区

6.11 mifare 卡操作函数

提供 MIFARE 卡读写接口

6.11.1 char mif_open(void);

//功能：打开并初始化读卡感应模块

6.11.2 char mif_close(void);

//功能：关闭读卡感应模块

6.11.3 char mif_request(unsigned char mode,unsigned char *atq);

功　　能：寻卡请求
入口参数：　　mode：寻卡模式
　　分为IDLE和ALL两种模式
　　0：表示为IDLE模式，一次只对一张卡操作
　　1：表示为ALL模式，一次可对多张卡操作
出口参数：atq：卡类型值
返 回：成功则返回 0

6.11.4 char mif_anticoll(unsigned char bcnt,unsigned char　*cardsnr);

功　　能：防止卡冲突，返回卡的序列号
入口参数：bcnt：预选卡所用的位数，标准值为0（不考虑系列号）
出口参数：cardsnr：返回的卡序列号地址
返　　回：成功则返回 0

6.11.5 char mif_select(unsigned char　*serial);

功　　能：从多个卡中选取一个给定序列号的卡
入口参数：serial：卡序列号
出口参数：
返　　回：成功则返回 0

6.11.6 char mif_load_key(unsigned char *uncodekey);

功　　能：将密码装入读写模块RAM中
入口参数：对于M1卡的每个扇区，密码包括A密码（KEYA）和B密码（KEYB）
　　uncodekey：写入RF MODLE的卡密码
返　　回：成功则返回 0

6.11.7 char mif_authentication(uchar auth_mode,uchar sactor,uchar *snr);

功　　能：验证某一扇区密码
入口参数：auth_mode：密码验证模式
　　sactor：要验证密码的扇区号（0～15）
返　　回：成功则返回 0

6.11.8 char mif_write(unsigned char blockaddr,unsigned char *W_data);

功　　能：向卡中写入数据
　　对于M1卡，一次必须写一个块，为16个字节；
入口参数：blockaddr：M1卡棗块地址（1～63）
　　W_Data：要写入的数据
返　　回：成功则返回0

6.11.9 char mif_read(unsigned char blockaddr,unsigned char *_data);

功　　能：读取卡中数据
　　对于M1卡，一次读一个块的数据，为16个字节；
入口参数：blockaddr：M1卡块地址（0～63）；
出口参数：_Data：读出数据
返　　回：成功则返回 0

6.11.10 char mif_increment(unsigned char block_addr, unsigned long value);

功　　能：块加值
入口参数：block_addr：块地址（1～63）
　　value：要增加的值
返　　回：成功则返回 0;

6.11.11 char mif_decrement(unsigned char block_addr, unsigned long value);

功　　能：块减值
入口参数：block_addr：块地址（1～63）
　　value：要减的值
返　　回：成功则返回 0

6.11.12 char mif_transfer(unsigned char block_addr);

功　　能：传送，将寄存器的内容传送到EEPROM中
入口参数：block_addr：要传送的地址（1～63）
返　　回：成功返回0

6.11.13 char mif_restore(unsigned char block_addr);

功　　能：回传函数，将EEPROM中的内容传入卡的内部寄存器
入口参数：block_addr：要进行回传的块地址（1～63）
返　　回：成功返回0

6.11.14 char mif_halt(void);

功　　能：中止卡操作
入口参数：
返　　回：成功则返回0

6.12 接触卡读写操作函数

手持机支持标准 ISO7816 接触卡读写,也支持如 4442,4428 卡读写，亦支持第二个 SAM 卡读写操作。

6.12.1 U8 ICC_IFOpen(U8 r);

功　　能： 打开接触卡模块
入口参数： r　=0
出口参数： 无
返　　回：成功则返回 0

6.12.2 U8 ICC_IFClose(void);

功　　能：　关闭接触卡模块
入口参数：　无
出口参数：　无
返　　回：成功则返回 0

6.12.3 U8 ICC_DetectCard(U8 Slot);

功　　能：　检测卡是否有效，仅对标准ISO大卡有效，SAM卡默认有卡存在
入口参数：　slot　选择卡座
　　1：选择机器内部第一个SAM卡座
　　2：选择大卡座或机器内部第二个SAM卡座
出口参数：　无
返　　回：=1有卡存在
　　=0无卡存在

6.12.4 U8 ICC_Reset(U8 Slot ,U8 *rlen,U8 *ATR);

功　　能：　标准ISO CPU卡复位命令
入口参数：　slot　　卡座选择
　　1：选择机器内部第一个SAM卡座
　　2：选择大卡座或机器内部第二个SAM卡座
出口参数：　*rlen　　复位字节长度
　　*ATR　复位返回的数据
返　　回：成功则返回 0

6.12.5 U8 ICC_PowerOpen(U8 Slot,U32 baud,U8 ucVoltage,U8 *ATR);

功　　能：　非标准CPU卡复位应答。
入口参数：　slot　　卡座选择
　　1：选择机器内部第一个SAM卡座
　　2：选择大卡座或机器内部第二个SAM卡座
　　baud　默认卡片速率
　　9600　与卡片通讯速率为9600　　bps
　　19200　与卡片通讯速率为19200　bps
　　38400　与卡片通讯速率为38400　bps
　　ucVoltage　默认卡片工作电压
　　1　　1.8V　工作电压
　　2　　3V　　工作电压
　　3　　5V　　工作电压
出口参数：　*ATR　复位返回的数据

返　　回：复位应答长度，=0 复位错误

6.12.6 U16 ICC_IsoCommand(U8 Slot,int slen, U8 *sbuf, U8 *rlen, U8 *rbuf);

功　　能：发送CPU卡APDU
入口参数：slot　卡座选择
　　1：选择机器内部第一个SAM卡座
　　2：选择大卡座或机器内部第二个SAM卡座
　　slen　　发送APDU长度
　　*sbuf　　发送数据缓冲区
出口参数：*rlen　　返回数据长度
　　*rbuf　　返回数据缓冲区
返　　回：返回CPU卡应答响应
　　0x6fff　命令执行错误 或CPU无响应

七、WINDOWS 平台通讯接口

此部分描述了在 WINDOWS 平台下与 PC 机通讯接口函数，通过这些接口，用户可以非常容易的将通讯接口连入系统管理，快速与手持机交换数据。我公司提供 VB,VF,PB,DELPHI,BCB 的接口例子，各种语言编程的具体调用请参考相关例子。

7.1、int openport(int port)；

作　　用：打开指定端口，并检测手持机是否连接在此端口上
调用参数：port 为当前手持机所连接的串口，如 1 为串口一，2 为串口二。
返回参数：成功　返回端口号（与 PORT 值相同），
　　失败　返回参数见错误列表。

7.2、void closeport(void)；

作　　用：关闭已打开的端口。
调用参数：无。
返回参数：无。
说　　明：与 openport 配合使用。

7.3、int downfile(int port,char *filename);

作　　用：从指定的端口下传一个名为[filename]的文件到手持机。
调用参数：port 为手持机连接的端口，
　　[filename]为一字符串变量，可以包含文件路径。
返回参数：成功　返回 0，
　　失败　返回参数见错误列表。
说　　明：此函数可以下装文件到手持机。

7.4、int upfile (int port,char *filename);

作　　用：从指定的端口上传一个名为[filename]的手持机文件到 PC
调用参数：port：为手持机连接的端口。
　　[filename]：为一字符串变量，因为手持机没有目录信息，因此[filename]不能包含文件路径。
返回参数：成功　返回 0
　　失败　返回参数见错误列表。
说　　明：在上传手持机文件之前，必须确定所要上传的文件存在于手持机里面，否则将返回一个错误。同时也要注意 PC 机的当前目录并没有此文件名，否则将被删除。

7.5、int findfile(int port,char *filename);

作　　用：查找一个名为[filename]的文件是否存在于手持机。
调用参数：port：为手持机连接的端口。
　　[filename]：为一字符串变量，因为手持机没有目录信息，因此[filename]不能包含文件路径。
返回参数：成功　返回 0 存在于手持机中
　　失败　返回参数见错误列表。

7.6、int deletefile(int port,char *filename);

作　　用：删除一个名为[filename]的手持机文件。
调用参数：port：为手持机连接的端口。
　　[filename]：为一字符串变量，因为手持机没有目录信息，因此[filename]不能包含文件路径。
返回参数：成功　返回 0
　　失败　返回参数见错误列表。

7.7、Int uptofile(int port,char *fname,char *fname2);

作　　用：上传并改文件名
调用参数：port：　　为手持机连接的端口
　　fname:　POS 机中的文件名
　　fname2:　PC 端的文件名，可以指定路径，也可以不指定路径，不指定时保存在当前目录
返回参数：成功　返回 0
　　失败　返回参数见错误列表。

7.8、int downtofile(int port,char *fname,char *filename);

作　　用：下传并改名
调用参数：　port：为手持机连接的端口
　　fname:　PC 端的文件名
　　fname2:　POS 机中的文件名
返回参数：成功　返回 0
　　失败　返回参数见错误列表。

7.9、int getlist(int port);

作　　用：返回文件列表信息。
调用参数：port：为手持机连接的端口。
返回参数：成功　返回文件个数 (>0 ,<128)
　　失败　返回相应错误码。
说　　明：　调用 getlist(port)函数之后，动态库会将手持机的文件分配表存储为当前目录中的 filetmp.tmp 临时文件中，文件名与文件大小放在此文件中，格式为文件长度[4]，文件名[11]。

文件头信息定义，		32BYTE
char dir_name[11];	文件名	11BYTE
U8 dir_attrib;	文件属性	1BYTE
U8 dir_case;	保留	1BYTE
U8 dir_crmonth;	月	1BYTE
U8 dir_crdate;	日	1BYTE
U16 dir_cryear;	年	2BYTE
U16 dir_start;	文件开始簇	2BYTE
U32 dir_size;	文件长度	4BYTE

7.10、int set_progress(int i_method,void* hwnd_proc);

作　　用：设置进度条方式。主要是提供给二次开发的用户以显示当前操作进度的

机会。具体说明如下：

调用参数：i_method：

0：返回 1。无进度显示；参数 2 无作用。

1：返回 1。 用户必须提供一个回调函数。此函数起到设置自己进度条进度的作用。在动态库执行过程中，会调用此函数，并传给此函数当前的进度(0—100)。参数 2 即为此函数的指针，此函数的调用习惯必须为 stdcall，原型如下：void set_prog_val(unsigned int ui_progress)。更具体的应用请参考我们提供的例子。

返回参数：成功 返回 0

失败 返回参数见错误列表。

说 明：i_method 参数为 1 时，回调函数必须有效，否则将出现非法操作，另外在此函数中不能出现耗时超过两秒的操作，例如 messagebox 等模态对话框。否则将阻塞整个通信进程导致超时错。

7.11、int downsystime (int port);

作 用：设置手持机日期和时间

调用参数：port：为手持机连接的端口。

返回参数：成功 返回 0

失败 返回参数见错误列表。

7.12、int getmno(int port,char *mno);

作 用：从指定的端口上传机器编号字符串到 PC。

调用参数：port：为手持机连接的端口。

[mno]：为一字符串变量,为返回的机器编号。

返回参数：成功 返回 0

失败 返回参数见错误列表。

说 明：如果你使用的编程语言对字符串的传递有困难的话，可以使用函数 int getnumfile(int port);此函数的作用与 getmno 一样，但没有字符串函数，函数将机器号保存到当前目录的一个名为 Machno 的文件中。

7.13、int getnumfile(int port);

作 用： 从指定的端口上传机器编号字符串到 PC,并保存在当前目录 Machno 文件中。

调用参数：port，为手持机连接的端口。

返回参数：成功 返回 0，

失败 返回参数见错误列表。

7.14、int down_userinfo (int port ,char *mno,char *mname);

作 用：更改手持机的操作员编号和姓名

调用参数：port，为手持机连接的端口。
Mno，操作员编号，最多 6 个字节
Mname,操作员姓名 最多 10 个字节
返回参数：成功 返回 0
失败 返回参数见错误列表
说 明：可以在 F2 菜单→系统信息→机器信息 可以看到下载的信息是否正确。

7.15、int up_userinfo (int port ,char *mno,char *mname);

作 用：上传手持机的操作员编号和姓名，
调用参数：port，为手持机连接的端口。
Mno，操作员编号，最多 6 个字节
Mname,操作员姓名 最多 10 个字节
返回参数：成功 返回 0
失败 返回参数见错误列表

7.16、int pos_exitcomm(port);

作 用：使手持机退出通讯状态
调用参数： port，为手持机连接的端口。
返回参数：成功 返回 0
失败 返回参数见错误列表

7.17 通讯返回值

返回值	含义
0	命令正常执行
101	无效数据
106	上装数据库成功
107	下装数据库成功
108	文件打开错误
109	内存错误
110	通信传输错误
111	通讯正常终止
112	通信接收错误
113	传输文件大于 64K
114	无效的文件尺寸
115	无效参数
116	下传程序完毕
117	删除完毕
118	串口设置完毕
119	手持机无响应
120	通信超时

123	文件格式错误
200	命令操作失败
201	文件未找到
202	文件删除失败
203	端口打开错误

附录一、 各键对应的 ASCII 码值

键名	键值	键名	键值	键名	键值	键名	键值
*	2AH	E	45H	Y	59h	r	72H
+	2BH	F	46H	Z	5AH	s	73H
–	2DH	G	47H	Λ	5EH	t	74H
.	2EH	H	48H	a	61H	u	75H
/	2FH	I	49H	b	62H	v	76H
0	30H	J	4AH	c	63H	w	77H
1	31H	K	4BH	d	64H	x	78H
2	32H	L	4CH	e	65H	y	79H
3	33H	M	4DH	f	66H	z	7AH
4	34H	N	4EH	g	67H	确认	8DH
5	35H	O	4FH	h	68H	运行	87H
6	36H	P	50H	i	69H	清除	82H
7	37H	Q	51H	j	6AH	F1	88H
8	38H	R	52H	k	6BH	F2	89H
9	39H	S	53H	l	6CH	▲	8EH
=	3DH	T	54H	m	6DH	▼	8FH
A	41H	U	55H	n	6EH	关	92H
B	42H	V	56H	o	6FH	空格	20H
C	43H	W	57H	p	70H		
D	44H	X	58H	q	71H		

附录二、系统保留宏定义

```
typedef unsigned char uchar;                //定义无符号数据    8bit
typedef unsigned short ushort;              //定义无符号数据    16bit
typedef unsigned long ulong;                //定义无符号数据    32bit
typedef  unsigned char U8;                  //定义无符号数据    8bit
typedef unsigned short U16;                 //定义无符号数据    16bit
typedef unsigned int U32;                   //定义无符号数据    32bit
typedef  signed char S8;                    //定义有符号数据    8bit
typedef signed short S16;                   //定义有符号数据    16bit
typedef signed int S32;                     //定义有符号数据    32bit
typedef unsigned char BYTE;                 //定义无符号数据    8bit
```

```
typedef  U8 *     P_U8;          /*   unsigned 8 bit data    */
typedef  U16 *    P_U16;         /*   unsigned 16 bit data */
typedef  U32 *    P_U32;         /*   unsigned 32 bit data */
typedef  S8 *     P_S8;          /*   signed 8 bit data    */
typedef  S16 *    P_S16;         /*   signed 16 bit data */
typedef  S32 *    P_S32;         /*   signed 32 bit data */
```

附录三、手持机在三表收费中的应用案例

手持机要配合电脑上的“用电（用水）收费软件”使用，根据各个用户的特点，手持机上的应用软件可能不同，所以需要注意三个方面：

A、 手持机中的数据库。

B、 手持机上的菜单功能和屏幕上需要显示的内容。

C、 电脑上用电（用水）收费软件的软件平台和数据库类型（如 DOS 或 WINDOWS）。

如果是我方工程师来开发此应用软件，则用户要向我方提供以上几个方面的信息。针对上述的 A、B、C 三个方面，下面详细阐述：

一、 针对 A，手持机上是以标准的数据库（*.dbf 格式）来存储的，我方手持机可以建立多个数据库，但大多数供电（供水）公司只需建立一个数据库，并通过此数据库来实现手持机和电脑之间的数据交换。抄表前电脑向手持机传送的数据库和抄完表后手持机向电脑传送的数据库是一样的数据库格式，但一些字段的内容已填满。根据我方的经验，一般常用的字段有：用户名、用户地址、用户号（或资产号、电脑号）、抄表标志（字符型）、上月读数、本月读数、上月电（水）量、本月电（水）量、倍率（管径）、用电（水）性质等。

如果条件允许，请提供下装手持机的样本数据库供我们参考。

对于电脑中用电收费软件其数据库不是 Fox 系列的*.dbf 格 式，则只需将手持机的数据库上传到电脑后，在电脑中利用相应的工具将数据库*.dbf 转换成用户需要类型的数据库格式。

二、针对 B，手持机抄表程序一般有以下几个菜单（或说几个功能）：

某某供电（供水）公司

1. 续上次抄表 注释：比如说，上次抄到第 7 户，这次则定位到第 8 户。
2. 查询抄表 如：输入用户号或电脑号，支持模糊查询和多号码查询。
3. 抄未抄户 自动搜索未抄的电表（水表）。
4. 统计 如：总户数，已抄户，未抄户，总电（水）量等.
5. 帮助 对手持机使用方法进行提示.

针对某一户电表，屏幕上显示的内容应根据我方手持机汉字 8 行 8 列（英文或数字是 16 行 16 列）的特点进行设计，一般显示内容如下：

用户名、用户地址、倍率（管径）、用电（水）性质、抄表标志、上月读数、本月读数、本月电（水）量等。

三、 针对 C，手持机同电脑通讯

如果电脑中是 DOS 环境，我方提供软件包*.exe 或*.plb 文件，其中*.plb 文件与*.prg 无区别，可以一起编译。如果是 WINDOWS 环境，我方提供动态连接库*.dll，此软件包简单易用，可在用户的用电管理程序之中象函数一样调用，而不影响管理系统的界面。为方便起见，我方还在 WINDOWS 下提供一个通讯安装程序。还提供 VF，PB，VB，C 等环境下的的通讯样板程序。

四、 在此提醒用户注意几个地方

（1）、我们一般在随机发货时会提供一套手持机“二次开发系统”工具包和一套“手持机同微机通讯”工具包，在软件包中，我方提供了两套手持机样板程序，此程序带有一定的代表性，我方并对此程序进行了详细的注释，用户的工程师也可参考一下。

（2）、在抄表时，当本月电（水）量是上月电（水）量的 2 倍或 1/2 时，手持机一般要提示抄表员，并弹出下列项供抄表员选择，1.未抄 2.正常 3.表快 4.表慢 5.表坏 6.估抄 7.其它 8.重抄，并将此代码存于“抄表标志字段”。当然，用户可以不要此提醒功能，或异常代码的名称也可能与上面所列的不相同。但“抄表标志”字段一定需要，为方便起见，一般初始值设为“1”。

注意： 如果贵公司对抄表异常情况有特殊处理要求的，请给出异常时的情况清单（必要时附代码）和详细的要求说明。

（3）、请一定写明手持机在每次抄表后需向电脑反馈信息的字段清单，即需要程序填补或更新内容的所有字段名。

（4）、对手持机中需要计算的字段，必须给出详细的计算公式和所有相关字段。

（5）、如果数据库中包含有除“抄表标志”以外的标志字段，请给出该字段的标志清单和初始值以及具体的判断方法。

（6）、如果贵公司有任何本表中未列出的其它特殊要求，请事先详加注明。

（7）、依次按“开”、“F1”、“0”，在屏幕下方显示的一串字符就是手持机的型号。在用户遇到一些问题时，可将此型号告诉我方工程师，便于迅速解决问题。

某某供电公司手持机应用程序开发要求
（电力范例）

一、数据库结构

字段序号	字段中文名	字段名	类型	长度	小数位数
1	电表号	DBH	N	8	
2	户名	HM	C	20	
3	用电性质	YDXZ	C	2	
4	上月行度	SYXD	N	8	
5	本月行度	BYXD	N	8	
6	抄表标志	CBBZ	N	2	
7	抄表日期	RQ	D	8	
8	用电地址	YDDZ	C	30	
9	倍率	BL	N	3	
10	抄表号	CBH	N	4	
11	上月电量	SYDL	N	10	
12	本月电量	BYDL	N	10	
13	新抄表号	XCBH	N	4	

二、手持机功能和屏幕显示内容

按[开]键开机后，再按[确认]键，运行抄表范例程序，屏幕显示：

```
        某某供电局
[1] 续上次抄表
[2] 抄未抄表
[F1]查询抄表
[3] 统计
[4] 帮助
[F2]退出
日期：   年 月 日
```

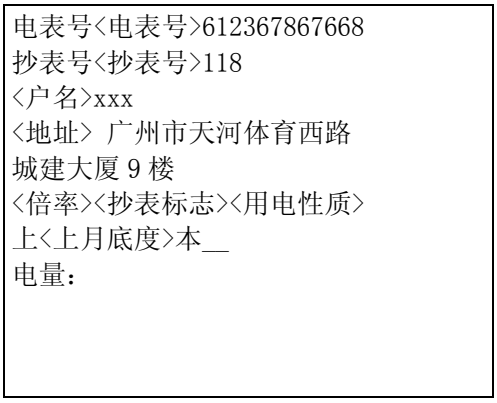
按照菜单提示，按数字键[1]至[4]，选择相应功能。

按[F1]键，可进入查询抄表状态。

按[F2]键，退出抄表管理程序，回到开机初始状态。此时可按[关]键关掉手持机。

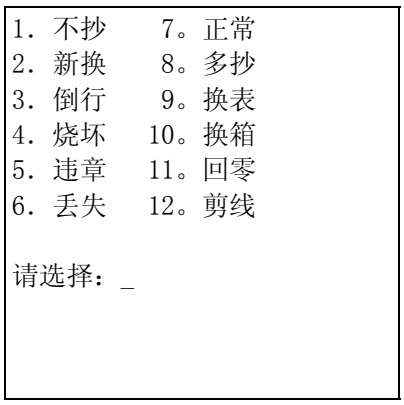
三、续上次抄表

自动定位到最后一次抄表退出前所指向的那条记录，屏幕显示如下:



（图 2）

“本”后出现光标，此时可输入本月底度，按 F1 也可进入查询抄表状态。按[清除]键可删除光标前一个字符，按[确认]健确认输入的本月底度后会自动计算出本月电量并显示出来。如果本月电量相比上月的电量没有什么问题，抄表标志显示为“已抄电表”，此时就抄完一块表。如果本月电量相比上月的电量小很多或大很多（一般超过 0. 5 或 2 倍）,此时标志显示为“异常”，异常时按[清除]键可重新输入数据，按其它任意键出现抄表标志选择菜单如下：



（图 3）

此时必须输入标志代码，输入后按[确认]键返回图（2）画面，此时也算抄完一块表。

当已经抄完一块表后，此时可按[F1]键直接进入查询抄表状态，按[清除]键清除本用户数据，置为未抄电表，可以重新抄表，按[∧]定位到上一条未抄记录，按[∨]或[确认]定位到下一条记录。不管该记录是否已抄.按[F2]可退回到主菜单。

四. 抄未抄表

从第一条记录开始搜索，直到找到第一条未抄电表的记录，并显示图（2），此时以下的操作方法基本同“续上次抄表”。所不同的是抄完一条记录之后按[∨]或[确认]将位到下一条未抄记录。

五、[F1]查询抄表

屏幕提示:“请输入电表或抄表号”，用户可以任意输入电表号或者是抄表号，并按[确认]键，若电表号或抄表号存在则定位到与之相符的那条记录，接着的抄表步骤同上。

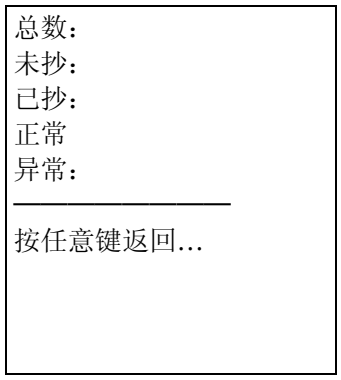
注意：

（1） 优先搜索电表号，如果没有找到相应的记录则手持机自动再搜索抄表号。

（2） 另外还提供模糊搜索，即：若输入电表号或抄表号不足十位，甚至更少，则手持机自动定位到前几位与之完全相符的第一条记录。

六. 抄表统计

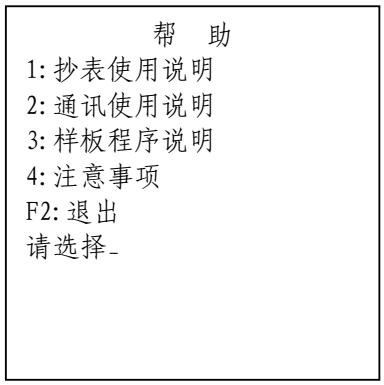
统计抄表情况，屏幕显示如下：



（图 4）

七. 帮助功能

提供程序和手持机的使用说明，注意事项；



（图 5）

选择相应的数字，则进入相应的帮助模块。在使用帮助功能时，按[∨]到下一页，[∧]键到上一页， [F2]退回到帮助菜单。

附录三、更改记录

- **v1.0 2005.10.11**
  初始版本
- **v2.0 2006.4.10**
  加入 Windows 平台动态库说明
- **v3.0 2006.8.3**
  加入打印函数接口
- **v4.0 2007.3.30**
  加入 MIFARE 卡接口
- **v4.2 2008.9.7**
  加入接触卡操作函数，对应机器版本：V06.02.84 以上