

# 目录

一、前言 .....	- 7 -
二、开发工具的安装和使用 .....	- 8 -
三、文件类 .....	- 9 -
1. int8 _Fopen(char *filename, char *mode) .....	- 9 -
2. Int8 _Fclose(int hdl).....	- 9 -
3. uint32 _Fread(void *ptr, uint32 size, int hdl).....	- 9 -
4. uint32 _Fwrite(void *ptr, uint32 size, int hdl) .....	- 9 -
5. int32 _Lseek(int hdl,uint32 offset,uint8 fromwhere) .....	- 10 -
6. uint32 _Filelenth(int hdl) .....	- 10 -
7. uint8 _Feof(int hdl) .....	- 10 -
8. int8 _Remove(char *filename).....	- 10 -
9. int8 _Access(const char *filepath, int mode);.....	- 10 -
10. int8 _MkDir(const char *path).....	- 10 -
11. int8 _ RmDir (const char *path).....	- 10 -
12. int32 _GetFreeSpace(void) .....	- 11 -
13. _dir * _OpenDir(const char *dirname) .....	- 11 -
14. _dirent * _ReadDir(_DIR *dirp).....	- 11 -
15. int _CloseDir(_DIR *dirp).....	- 11 -
16. int _Rename(const char *oldPath, const char *newPath) .....	- 11 -
17. int32 _GetFileAtt(const char *Path,_stat *stat) .....	- 11 -
18. char * GetFileList(char *title,char *ext,char *dir).....	- 11 -
四、数据库类 .....	- 11 -
1. void _Select(int dbfno ) .....	- 11 -
2. void _Go(int recno).....	- 12 -
3. void _Skip(int recno) .....	- 12 -
4. int _Use(char *dbffile).....	- 12 -
5. void _ReadField(uint8 dbfoffset,char *field).....	- 12 -
6. uint32 _Recno(void) .....	- 12 -
7. uint32 _Reccount(void) .....	- 12 -
8. uint8 _Bof(void) .....	- 12 -
9. uint8 _Eof(void).....	- 13 -
10. void _Replace (uint8 dbfoffset,char *field) .....	- 13 -
11. void _App (void).....	- 13 -
12. void _Del (void).....	- 13 -
13. void _Pack(void).....	- 13 -
14. void _Zap(void) .....	- 13 -
15. int _Locate(uint8 FieldName,char condition,char Contect[],int32 RecStart,int32	

RecEnd,int fx).....	- 13 -
16. void _GetFieldStr(uint8 fieldname, struct _Field *dbfstr).....	- 14 -
17. void _Deleted(void).....	- 14 -
-	
18. char * _ReadField(uint8 dbfoffset,char *field).....	- 14 -
-	

## 五、时间类 ..... - 14 -

1. void _GetTime(char *time,char div).....	- 14 -
2. void _GetDate(char *date,char div) .....	- 14 -
3. void _SetTime(char *time) .....	- 15 -
4. void _SetDate(char *date).....	- 15 -
5. uint8 _GetWeek(void).....	- 15 -
6. uint8 _GetDay(void) .....	- 15 -
7. uint8 _GetMonth(void) .....	- 15 -
8. uint32 _GetYear(void) .....	- 15 -
9. void _GetDateTime(char *time,char time1,char time2) .....	- 15 -
10. void _SetDateTime(char *datetime) .....	- 15 -
11. uint8 _GetSec(void) .....	- 16 -
12. uint8 _GetMin(void) .....	- 16 -
13. uint8 _GetHour(void).....	- 16 -
14. volatile uint32 _GetTickCount(void) .....	- 16 -
15. void _Sleep(uint32).....	- 16 -

## 六、液晶显示类 ..... - 16 -

1. void _ ClearScreen (void) .....	- 16 -
2. void _Printfxy(int x,int y, char *str,int color) .....	- 17 -
3. void _LCMXorDataXY(uint8 x, uint8 y,uint8 x1,uint8 y1) .....	- 17 -
4. void _GUIHLine(uint8 x0, uint8 y0, uint8 x1, uint8 color).....	- 17 -
5. void _GUIVLine(uint8 x0, uint8 y0, uint8 y1, uint8 color).....	- 17 -
6. void _GUILine(uint32 x0, uint32 y0, uint32 x1, uint32 y1, uint8 color) .....	- 17 -
7. void _GUIRectangle(uint32 x0 , uint32 y0, uint32 x1, uint32 y1 , uint8 color) .....	- 17 -
8. void _GUIRectangleFill(uint32 x, uint32 y, uint32 x1, uint32 y1, uint8 color) .....	- 17 -
9. void _GUIPoint(uint8 x, uint8 y, uint8 color).....	- 18 -
10. void _SaveScreenToBuff(uint8 *buff) .....	- 18 -
11. void _RestoreBuffToScreen(uint8 *buff).....	- 18 -
12. void _Printfxy12(int x,int y, char *str,int).....	- 18 -
13. void _GUILoadPic(_PicStru *).....	- 18 -
14. char _T9PY__Get__Match__PY__MB(char *p_PadInput,const struct t9PY_index ** List_match_PY_Mb).....	- 18 -
15. void _toxy(unsigned char X,unsigned char Y) .....	- 18 -
16. void _HiderCur(void).....	- 18 -
17. void _ ShowCur(void).....	- 18 -
18. void _GetCursor(_uint8 *,uint8 *).....	- 18 -

## 七、液晶类组件 . . . . . - 19 -

1. uint8 _MessageBox(_GuiStrStru *stru).....	- 19 -
2. uint8 _Menu(_GuiMenuStru* menu).....	- 20 -
3. uint8 _MenuEx(_GuiMenuStru * menu) .....	- 20 -
4. uint8 _List(_GuiLisStru t * list).....	- 20 -
5. uint8 _InputBox(_GuiInputBoxStru *,inputbox).....	- 21 -
6. Uint8 _GetStr(_GuiInputBoxStru *).....	- 21 -
7. Void _SetInputMode (uint32) .....	- 21 -
8. Void _DisInputMode (uint32).....	- 21 -
9. Uint8 _MessageBoxEx (char *lpCaption ,char *str,, uint8 itype) .....	- 22 -
10. uint32_ListEx(_GuiListStruEx *).....	- 18 -
11. uint8 _CreateProgressBar(_ProGressBar *)......	- 18 -
12. uint8 _IncrementProgressBar(_ProGressBar*).....	- 18 -
13. void _SetFrontSize(uint8 size).....	- 18 -

## 八、串口类 . . . . . - 22 -

1. void _ComSetTran(uint8 Tran) .....	- 23 -
2. void _ComSet(uint8 *Setting,uint8 comport) .....	- 23 -
3. void _SendComStr(uint8 *SendStr,uint32 SendSize).....	- 23 -
4. uint32 _GetComStr(uint8* Getstr,uint32 GetSize,uint32 GetMs) .....	- 23 -
5. void _CloseCom(void).....	- 23 -

## 九、按键类 . . . . . - 23 -

1. uint32 _ReadKey(void).....	- 23 -
2. uint32 _GetKeyExt(void).....	- 24 -
3. void _SetKey(uint8 ROW_KEY_LOG) .....	- 24 -

## 十、控制类 . . . . . - 24 -

1. void _OpenRedLight(void).....	- 24 -
2. void _CloseRedLight(void) .....	- 24 -
3. void _OpenGreenLight(void).....	- 24 -
4. void _CloseGreenLight(void) .....	- 25 -
5. uint32 _GetShutDonwTime(void) .....	- 25 -
6. void _SetShutDonwTime(uint32 time).....	- 25 -
7. void _SetBeepOnOff(uint8 onoff) .....	- 25 -

8. uint32 _GetMbt(void) .....	- 25 -
9. uint32 _GetSbt(void) .....	- 25 -
10. void _Beep(void) .....	- 25 -
11. void _OpenLcdBackLight(void) .....	- 25 -
12. void _CloseLcdBackLight(void).....	- 25 -
13. void _OpenKeyBackLight(void).....	- 26 -
14. void _CloseKeyBackLight(void) .....	- 26 -
15. void _EnableArmIrqInt(void) .....	- 26 -
16. void _DisableArmIrqInt (void) .....	- 26 -
17. void _SetBeepOnOff(uint8 onoff) .....	- 18 -
18. void _OpenFlashLight(void).....	- 18 -
19. void _CloseFlashLight(void) .....	- 18 -

## 十一、系统菜单类 ..... - 26 -

1. void _SetBacklightSys (void) .....	- 26 -
2. void _SetLCDContrastSys (void) .....	- 26 -
3. void _ShutDownPower(void) .....	- 26 -
4. void _CalendarSys(void) .....	- 27 -
5. void _BatteryDisplaySys(void).....	- 27 -
6. void _CalculatorSys(void) .....	- 27 -
7. void _SoundSwitchSys(void).....	- 27 -
8. void _FileTransSys(void).....	- 27 -
9. void _FormatSys(void) .....	- 27 -
10. void _AutoShutDonwMenuSys(void).....	- 27 -
11. void _TimeSettingSys(void) .....	- 27 -
12. void _InfoSys(void) .....	- 27 -
13. void _DispTimeSys(void) .....	- 28 -
14. void _GetMachineSn(char *) .....	- 28 -
15. void _SetComSever(void).....	- 28 -
16. void _Reset(void).....	- 28 -
17. void _SetFileTran(char flag,char btl).....	- 28 -
18. void _GetTransCommPort (uint32 *COM,uint32 *BTL).....	- 28 -
19. void _SetShutDownTimeEx (uint32).....	- 28 -
20. void _SoundOn(void).....	- 18 -
21. void _SoundOff(void).....	- 18 -
22. void _RunHex(char *filename).....	- 18 -
23. void _IsPowerKey(uint8).....	- 18 -

## 十二、RFID 类..... - 29 -

1. void _RfidInit(void).....	- 29 -
2. uint16 _RfidFind(uint8 active) .....	- 29 -
3. uint16 _RfidReadCardId(uint8 cardid[]) .....	- 29 -
4. uint16 _RfidSelectCard(uint8 cardid[]) .....	- 29 -
5. uint16 _RfidCheckPwd(uint8 pwd[],uint8 cardid[],uint8 address,uint8 type) .	- 30 -
6. uint16 _RfidWriteData(uint8 data[],uint8 address).....	- 30 -
7. uint16 _RfidReadData(uint8 data[],uint8 address) .....	- 30 -
8. uint16 _RfidCharge(uint32 value,uint8 address,uint8 type) .....	- 30 -
9. uint16 _RfidBackup(uint8 address) .....	- 30 -
10. uint16 _RfidDormancy(void).....	- 31 -
11. uint16 _RfidReset(void).....	- 31 -
12. uint16 _RfidOpenAntenna(void) .....	- 31 -- 31 -
13. uint16 _RfidCloseAntenna(void).....	- 31 -
14. void _RfidClose(void) .....	- 31 -
15. uint16 _RfidInitWallet(uint32 value, uint8 address).....	- 18 -
16. uint32 _RfidReadWallet(uint8 daaress) .....	- 18 -
17. int8 _Rfid125Read(uint8 *id, int32 len, int32 ms) .....	- 18 -
18. int _Rfid125Start(void) .....	- 18 -
19. void _Rfid125End(void) .....	- 18 -

### 十三、扫描条形码类 . . . . . - 32 -

1. Unit8 _ScanBarcodeKey(unit8 *barcode,uint8 evkey).....	- 32 -
2. Uint8 _ScanBarcode(uint8 *barcode).....	- 32 -

### 十四、蓝牙函数（A380 机器） . . . . . - 32 -

1. unit8 BtOpen(void).....	- 32 -
2. void BtClose(void).....	- 32 -
3. void BtSetMode(unit8 atmdel).....	- 33 -
4. unit8 BtGetModel(void).....	- 33 -
5. int8 _BtMrad(char *addr) .....	- 18 -
6. int8 _BtFsad(char *addr) .....	- 18 -
7. void _BtGetList(void).....	- 18 -

返回值：无 .....	- 33 -
8.Int8 BtScan(uint8 num.char *cClass.char *clac) .....	- 33 -
9. Int8 BtName(char *name,uint8 op) .....	- 33 -
10. int8 _BtRName(char*name ,char *addr) .....	- 33 -
11. int8 _BtAddr(char *addr).....	- 34 -
12. int8 _BtPair(char *addr,uint16 ms).....	- 34 -
13. int8 _BtLink(char *addr) .....	- 34 -
14. int8 _BtPswd(char *pswd,uint8 op).....	- 34 -
15. uint32 _BtGetData(uint8 *RecBuff,uint32 len,uint16 ms) .....	- 35 -
16. int8 _BtDefault(void).....	- 35 -

17. int16 _BtSendData(uint8 * Buff,uint32 Len).....	- 35 -
18. int8 _BtSetRole(uint8 role).....	- 35 -

## 十五、WIFI 函数(A380 机器) . . . . . - 36 -

1. int8 _WifiAbrd(void) .....	- 36 -
2. int8 _WifiUpgrade(void).....	- 36 -
3. int8 _WifiLoadDone (void).....	- 37 -
4. int8 _WifiBand(uint8 band_val) .....	- 37 -
5. int8 _WifiInit(void).....	- 37 -
6.int8 _WifiPassscan(uint16 bit_map) .....	- 37 -
7. int8 _WifiSetNetworkType(char *Type).....	- 38 -
8. int8 _WifiPreSharedKey(char *psk) .....	- 38 -
9. int8 _WifiScan(uint8 chan_num,char *ssid, Access_Points *ap,uint8 ap_num) .....	- 38 -
10.int8 _WifiAuthmode(uint8 value).....	- 39 -
11. int8 _WifiJoin(char *ssid_name,uint8 TxRate,uint8 TxPower).....	- 39 -
12.int8 _WifiSetIp(uint8 dhcp,char *ip,char *subnet,char *gateway) .....	- 40 -
13. int8 _WifiCreateSocket(int8 type ,char *ipaddr,char *portno,char *lport).....	- 40 -
14. int8 _WifiCloseSocket(int8 socketno) .....	- 41 -
15. int16 _WifiSendData(int8 sockno,uint16 slength,char *sdata,char *dip,char *dport). -	41 -
16. int16 _WifiReceiveData(int8 *sockno,char *sdata,uint16 ms).....	- 41 -
17. int8 _WifiClose(void) .....	- 41 -
18. int8 _WifiOpen(void).....	- 18 -

## 十六、GPRS 函数（A380 机器） . . . . . - 42 -

1. int16 _GprsGetSignal(char *cResult) .....	- 42 -
2. int16 _GprsGetRegistration(char *cResult) .....	- 42 -
3. int16 _GprsGetCurrentSelection(char *cResult) .....	- 42 -
4. int16 _GprsGetAllSelection(char *cResult) .....	- 42 -
5. int16 _GprsGetSimPinStatus(char *cResult).....	- 43 -
6. int16 _GprsSetSimPin(char *cPin,char *cNewPin).....	- 42 -
7. int16 _GprsSetService(uint8 uOperate) .....	- 43 -
8. int16 _GprsGetService(char *cResult) .....	- 43 -
9. int16 _GprsSetMultiIp(uint8 uOperate).....	- 43 -
10. int16 _GprsGetMultiIp(void).....	- 44 -
11. int16 _GprsStartTask(char *cApn,char *cName,char *cPassword).....	- 44 -
12. int16 _GprsActivatePDP(void) .....	- 44 -
13. int16 _GprsGetLocalIp(char *ip).....	- 44 -
14. int16 _GprsSetConnection(char*cMode,char *cIp, char * cPort,uint16 ms).....	- 45 -
15. int16 _GprsGetIpStatus(int8 iMultiIp,char *cStatus) .....	- 45 -
16. int16 _GprsSendData(uint8 *cData,uint16 iLength) .....	- 46 -
17. int16 _GprsIpClose(void) .....	- 46 -
18. int16 _GprsDeactivatePDP(void) .....	- 46 -
19. int16 _GprsSetTcpipMode(uint8 uMode) .....	- 46 -

20. int16 _GprsGetTcpipMode(void).....	- 46 -
21. int16 _GprsGetAtStatus(void) .....	- 46 -
22. int16 _GprsSetDefault(void).....	- 47 -
23. int8 _GprsOpen(uint32 iBaud) .....	- 47 -
24. int8 _GprsClose(void).....	- 47 -
25.int16 _GprsSetSingleConnection(char *cApn,char*cMode,char *cIp, char* iPort) ....	- 47 -
<b>十七、其他类</b> .....	<b>- 47 -</b>
1. void _DoubleToStr(char *s1 , double s2 ,int n ) .....	- 48 -
2. void _GetSysVer(char * ver,char *UpdateTime) .....	- 48 -
<b>十八：C 语言函数标准函数(参考 C 语言手册)</b> .....	<b>- 48 -</b>
<b>十九、与电脑接口动态库说明</b> .....	<b>- 49 -</b>
数据下装 .....	- 49 -
数据上装 .....	- 50 -
获取抄表机序列号 .....	- 51 -
设置抄表机系统时间 .....	- 52 -
获取机器里面目录 .....	- 52 -
建目录 .....	- 53 -
删除目录 .....	- 54 -
删除文件 .....	- 54 -
<b>板上接口</b> .....	<b>- 56 -</b>

## 一、前言

本使用手册详细说明了数据采集终端二次开发工具，开发流程，API 函数请在使用前仔细阅读该使用手册，以便保证您的正确使用和工作效率，敬请妥善保管以备日后查阅之用。如发现本手册有编写错误之处，请予以指出，以便更正，谢谢！

## 二、开发工具的安装和使用

**安装工具：**从用户光盘中直接拷贝 ARM 目录到 C 盘,该目录含有编译工具,库函数.

**编译工具配置：**我们建议使用 EditPlus v3.20 ,把用户光盘中的 \编程工具\EditPlus 素材\所有文件拷贝到安装目录下。然后运行，EditPlus 在菜单中的工具中选择配置用户工具，在工具 1 中点击添加工具->应用程序 填写下面内容

菜单文本： 编译

命令： C: \arm\arm\HJ.exe

参数： \$(FilePath) c: \arm\include c: \arm\lib

然后把捕捉输出选项勾上，点击输出模式：把使用默认输出模式去掉；正则表达式为：  
^[^]+ (.+) ([0-9]+ :); 文件名： 无,行： 标记表达式 2;列 无

再次点击添加工具->应用程序 填写下面内容

菜单文本： 下载

命令： 在 EditPlus 里面选择 DOWM.EXE

参数： 0 \$(FilePath)

然后把捕捉输出选项勾上(其中 0 是代表 COM1)

如上述增加

菜单文本： 转换

命令： 在 EditPlus 里面选择 DBF.EXE

参数： \$(FilePath)

然后把捕捉输出选项勾上

在 EditPlus 在菜单中的工具中\用户工具组选择 用户工作组 1，那么在当前工具栏上会显示“编译 CTRL+1”，下载程序 CTRL+2 ”DBF 转换 CTRL+3.”

在菜单中的工具中选择用户工作组->工具组 1。编译的时候我们打开一个 C 语言的代码，快捷键用法：

编译 CTRL+1： 就可以编译,如果有错误提示,点击下面的提示错误信息,就可以自动转到错误的地方.

下载 CTRL+2： 自动下载源代码下文件夹里的\*.HEX 文件到机器上（机器处在数据通信状态下）。

转换 CTRL+3： 如果你是对 DBF 操作，它会在你的程序目录找 DBF 文件，而且自动产生 DBFH 文件(该文件就是数据库里面的字段的定义)。在素材窗体里面右键点击重新载入，数据库字段就会显示在素材中的“DBF 字段数据“的目录下。点击该字段，那么该字段直接显示在程序代码里面。另外在菜单中的文档\文件类型点击 刷新 STX/ACP(R),就可以在程序中看到 DBF 字段的颜色产生变化。

## 二次开发库说明



如无特殊说明,所有函数名字都可以在 C: \arm\Include\ hplib.h 里面可以找到.  
版本说明: V1.01

### 三、文件类

#### 1. int8 \_Fopen(char \*filename, char \*mode)

功能说明: 打开一个文件

参数说明:

Filename: 文件名字

Mode: 'R' 只读,'RW',读写,'W' 创建文件

返回值: -1 打开失败, 其他成功

#### 2. Int8 \_Fclose(int hdl)

功能说明: 关闭文件

参数说明:

hdl: 文件句柄

返回值: -1 关闭失败, 0 成功

#### 3. uint32 \_Fread(void \*ptr, uint32 size, int hdl)

功能说明: 读文件

参数说明:

ptr: 指向要读进来的缓冲区

size: 要读取内容的长度

返回值: 读取文件具体字节

#### 4. uint32 \_Fwrite(void \*ptr, uint32 size, int hdl)

功能说明: 写文件

参数说明: 如果 ptr 和 size 都为 0, 则在当前位置截断文件

ptr: 指向要写进来的缓冲区

size: 要写的内容的长度

hdl: 文件句柄

返回值: 写进文件具体字节

## 5. int32 \_Lseek(int hdl,uint32 offset,uint8 fromwhere)

功能说明：文件指针移动

参数说明：

hdl：文件句柄

offset：移动的长度

fromwhere：从那里移动 0 文件开始移动,1 文件当前移动,2 文件末尾向前移动

返回值：移动长度，-1 操作失败，

## 6. uint32 \_Filelenth(int hdl)

功能说明：获取文件大小

参数说明：

hdl：文件句柄

返回值：文件大小

## 7. uint8 \_Feof(int hdl)

功能说明：判断是非到文件结尾

参数说明：

hdl：文件句柄

返回值：1 文件结尾

## 8. int8 \_Remove(char \*filename)

功能说明：删除文件

参数说明：

filename：文件名

返回值：-1 失败，0 成功

## 9. int8 \_Access(const char \*filenpath, int mode);

功能说明：检查文件

参数说明：filenpath 文件名字

mode：为 0

返回值：0 存在，-1 不存在

## 10. int8 \_MkDir(const char \*path)

功能说明：建立目录

参数说明：path 为目录，如 test 建立 test 目录

mode：为 0

返回值：0 成功，-1 失败

## 11. int8 \_Rmdir(const char \*path)

功能说明：删除目录

参数说明：path 为目录

mode：为 0

返回值：0 成功，-1 失败

12. int32 \_GetFreeSpace(void)

功能说明：获取剩余磁盘空间

参数说明：

返回值： 磁盘剩余空间

13. \_dir \* \_OpenDir(const char \*dirname)

功能说明：打开目录

参数说明： dirname 目录

返回值： yaffs\_DIR 结构体

14. \_dirent \* \_ReadDir(\_DIR \*dirp)

功能说明 读取信息

参数说明： DIR 结构体

返回值： \_diren 结构体

15. int \_CloseDir(\_DIR \*dirp)

功能说明 关闭目录

参数说明： DIR 结构体

返回值：

16. int \_Rename(const char \*oldPath, const char \*newPath)

功能说明： 文件重命名

参数说明： oldPath 原来文件名， newPath 新的文件名

返回值： -1 失败， 0 成功

17. int32 \_GetFileAtt(const char \*Path,\_stat \*stat)

功能说明：获取指定目录/文件属性

参数说明： Path 目录/文件， stat 属性

返回值： -1 失败， 0 成功

18. char \* GetFileList(char \*title,char \*ext,char \*dir)

功能说明：列表方式选择文件

参数说明： title 为显示标题， ext 为文件扩展名(NULL 是所有文件),dir 为目录

Ext : NULL 所有文件

dir : NULL 根目录

返回值： NULL 取消选择，非空，指向文件名的指针

## 四、数据库类

1. void \_Select(int dbfno )

功能说明：选定一个工作区为当前工作区

参数说明：

dbfno: 数据库区

返回值：无

## 2. void \_Go(int recno)

功能说明：定位记录

参数说明：

recno: 记录号

返回值：无

## 3. void \_Skip(int recno)

功能说明：相对移动记录

参数说明：

recno: 记录号 正往下移动,负往上移动

返回值：无

## 4. Int \_Use(char \*dbffile)

功能说明：打开数据库

参数说明：

dbffile: 文件名,如果是空则为关闭数据库

返回值：成功返回 1，失败返回 0;

## 5. void \_ReadField(uint8 dbffoffset,char \*field)

功能说明：读字段

参数说明：

dbffoffset: 字段偏移量,如果是一个字段则为 0

field: 要读入的缓冲区

返回值：无

## 6. uint32 \_Recno(void)

功能说明：返回当前记录号

参数说明：无

返回值：当前记录号

## 7. uint32 \_Reccount(void)

功能说明：数据库总记录

参数说明：无

返回值：总记录

## 8. uint8 \_Bof(void)

功能说明：检测是否到第一条记录

参数说明：无

返回值： 1 代表当前是第一条记录,0 不是第一条记录

#### 9. uint8 \_Eof(void)

功能说明： 检测是否到最后记录

参数说明： 无

返回值： 1 代表最后的记录,0 不是最后的记录

#### 10. void \_Replace (uint8 dbfoffset,char \*field)

功能说明： 字段更新

参数说明：

Dbfoffset : 字段偏移量

Field : 要更新的内容

返回值 : 无

#### 11. void \_App (void)

功能说明： 追加一个空记录

参数说明： 无

返回值 : 无

#### 12. void \_Del (void)

功能说明： 标记当前记录已被删除(逻辑删除)

参数说明： 无

返回值： 无

#### 13. void \_Pack(void)

功能说明： 物理删除已标记删除的记录

参数说明： 无

返回值： 无

#### 14. void \_Zap(void)

功能说明： 清空数据库

参数说明： 无

返回值： 无

#### 15. void int \_Locate(uint8 FieldName,char condition,char Contect[],int32 RecStart,int32 RecEnd,int fx)

功能说明： 查找记录

参数说明： FieldName 字段名字

Condition 查找条件

Contect 匹配内容

RecStart 开始记录

RecEnd 结束记录

Fx 0 精确,1 模糊,2,前模糊,3 后模糊

返回值： 1 查找成功，直接定位到目标记录，

0 查找失败，返回第一条的开始查的记录

```
extern int    _Locate
(uint8 FieldName,char condition,char Conect[],int32 RecStart,int32 RecEnd,int fn); //Fn=0 精
确,1 模糊,2,前模糊,3 后模糊
```

#### 16. void \_GetFieldStr(uint8 fieldname, struct \_Field \*dbfstr)

功能说明： 获取字段属性

参数说明：

Fieldname: 字段名称

```
struct _Field{
    uint16  offset;
    uint16  len;
    uint8   type;
    uint8   adix  ;
```

```
};
```

返回值： 无

#### 17. Uint8 \_Deleted(void)

功能说明： 判断当前记录是否被标记为删除

参数说明： 无

返回值： 0--未标记为删除 ， 1--标记为删除

#### 18. void \_ReadFieldEx(uint8 dbffoffset,char \*field)

功能说明： 读取字段，且不会省略字段中的空格

参数说明：

uint8 dbffoffset: 字段偏移量,如果是一个字段则为 0

Char \*field: 要读入的缓冲区

返回值： 无

## 五、时间类

#### 1. void \_GetTime(char \*time,char div)

功能说明： 获取系统时间

参数说明：

time: 时间缓冲区

div: 分隔符 如':',''

返回值： 无

#### 2. void \_GetDate(char \*date,char div)

功能说明： 获取系统日期

参数说明：

date: 日期缓冲区

div: 分隔符 如'/','-',''

返回值： 无

3. void \_SetTime(char \*time)  
功能说明： 设置系统时间  
参数说明：  
    time: 指定时间 如'12: 12: 12'  
返回值： 无
4. void \_SetDate(char \*date)  
功能说明： 设置系统日期  
参数说明：  
    date: 指定时间 如'2012-12-12'  
返回值： 无
5. uint8 \_GetWeek(void)  
功能说明： 获取系统周  
参数说明： 无  
返回值： 周
6. uint8 \_GetDay(void)  
功能说明： 获取系统日期号  
参数说明： 无  
返回值： 号
7. uint8 \_GetMonth(void)  
功能说明： 获取系统月份  
参数说明： 无  
返回值： 月份
8. uint32 \_GetYear(void)  
功能说明： 获取系统年份  
参数说明： 无  
返回值： 年份
9. void \_GetDateTime(char \*time,char time1,char time2)  
功能说明： 获取系统日期时间  
参数说明：  
    time: 日期时间缓冲区  
    time1: 日期分隔符,  
    time2: 时间分隔符  
返回值： 无
10. void \_SetDateTime(char \*datetime)  
功能说明： 设置系统日期时间  
参数说明：  
    time: 指定日期时间 如'2010 -10-10 10-10-10'

返回值：无

#### 11. uint8 \_GetSec(void)

功能说明：获取秒

参数说明：无

返回值：秒

#### 12. uint8 \_GetMin(void)

功能说明：获取分

参数说明：无

返回值：分钟

#### 13. uint8 \_GetHour(void)

功能说明：获取时

参数说明：无

返回值：小时

#### 14. volatile uint32 \_GetTickCount(void)

功能说明：读定时器数值

参数说明：读取实时时间  $1 = 1/32768$  秒

返回值：0—1966079

#### 15. void \_Sleep(uint32)

功能说明：休眠等待

参数说明：无

返回值：小时

## 六、液晶显示类

```
typedef struct {  
    int x; 开始点 X 坐标  
    int y; 开始点 Y 坐标  
    int x1; 结束点 X1 坐标  
    int y1; 结束点 X1 坐标  
    uint8 color; 颜色  
    char *str; 具体内容  
    char *lpCaption; 显示标题  
    uint8 itype; 内容  
}GUI_STR;
```

#### 1. void \_ClearScreen (void)

功能说明：清除屏幕显示

参数说明：无



返回值：无

2. void \_Printfxy(int x,int y, char \*str,int color)

功能说明：在指定的点上显示字符串

参数说明：x,y 坐标，color 0 白色，1 黑色

str：要显示的字符串

返回值：无

3. void \_LCMXorDataXY(uint8 x, uint8 y,uint8 x1,uint8 y1)

功能说明：反色区域

参数说明：

x,y,x1,y1,坐标

返回值：无

4. void \_GUIHLine(uint8 x0, uint8 y0, uint8 x1, uint8 color)

功能说明：画横线

参数说明：

x,y,x1,坐标

color 颜色, 0,为白色,1 为黑色

返回值：无

5. void \_GUIVLine(uint8 x0, uint8 y0, uint8 y1, uint8 color)

功能说明：画竖线

参数说明：

x,y,y1,坐标

color 颜色, 0,为白色,1 为黑色

返回值：无

6. void \_GUILine(uint32 x0, uint32 y0, uint32 x1, uint32 y1, uint8 color)

功能说明：画线

参数说明：

x,y,x1,y1,坐标

color 颜色, 0,为白色,1 为黑色

返回值：无

7. void \_GUIRectangle(uint32 x0 , uint32 y0, uint32 x1, uint32 y1 , uint8 color)

功能说明：画矩形

参数说明：

x,y,x1,x1,坐标

color 颜色, 0,为白色,1 为黑色

返回值：无

8. void \_GUIRectangleFill(uint32 x, uint32 y, uint32 x1, uint32 y1, uint8 color)

功能说明：填充区域

参数说明:

x,y,x1,y1,坐标

color 颜色, 0,为白色,1 为黑色

返回值: 无

#### 9. void \_GUIPoint(uint8 x, uint8 y, uint8 color)

功能说明: 画点

参数说明:

x,y 坐标

color 颜色, 0,为白色,1 为黑色

#### 10. void \_SaveScreenToBuff(uint8 \*buff)

功能说明: 保存屏幕

参数说明: 缓冲区

返回值: 无

#### 11. void \_RestoreBuffToScreen(uint8 \*buff)

功能说明: 保存屏幕

参数说明: 缓冲区

返回值: 无

#### 12. void \_Printfxy12(int x,int y, char \*str,int color)

功能说明: 显示 12 点阵 (0~9)

参数说明:

x,y 坐标

color 0 白色, 1 黑色

str : 只能显示 12 点阵 0—9

返回值: 无

#### 13. void \_GUILoadPic(\_PicStru \*)

功能说明: 在指定的上显示图片

参数说明:

stru:

```
typedef struct {  
    uint32 x; X 坐标  
    uint32 y; Y 坐标  
    uint8 *dat; 图片内容  
    uint32 hno ; 行数  
    uint32 lno; 列数  
    uint8 diff; 0  
}_PicStru
```

#### 14. char \_T9PY\_Get\_Match\_PY\_MB(char \*p\_PadInput, const struct t9PY\_index \*\*

List\_match\_PY\_Mb)

功能说明:

参数说明:

Char \*p\_PadInput:

Const struct t9PY\_index \*\*List\_match\_PY\_Mb:

```
struct t9PY_index
{
    char    *t9PY_T9;
    char    *PY;
    const char    *PY_mb;
};
```

返回值:

15. void \_toxy(unsigned char X,unsigned char Y)

功能说明: 将光标定位到具体位置

参数说明:

unsigned char X: X 座标

unsigned char Y: Y 座标

返回值:

16. void \_HideCur(void)

功能说明: 隐藏光标

参数说明: 无

返回值: 无

17. void \_ShowCur(void)

功能说明: 显示光标

参数说明: 无

返回值: 无

18. void \_GetCursor(uint8\*,uint\*8)

功能说明: 获取光标位置

参数说明:

返回值: 无

## 七、液晶类组件

1. uint8 \_MessageBox(\_GuiStrStru \*stru)

功能说明: 提示框

参数说明:

stru:

```
typedef struct {
    char *str;          具体内容
    char *lpCaption;    显示标题
    uint8 itype;        0 表示只有确定,1 表示显示确定取消
} _GuiStrStru;
```

返回值: IDOK(1), IDCANCEL(2)

## 2. uint8 \_Menu(\_GuiMenuStru\* menu)

功能说明: 菜单显示

参数说明:

menu: 只有退出键才能退出

```
typedef struct
{
    WINDOWS *win;
    uint32 left;          窗口位置(左上角的 x 坐标)
    uint32 top;
    char *title;          菜单标题
    uint8 no;             主菜单个数<=8
    char *str[MMENU_NO];  主菜单字符串
    char *key[MMENU_NO];  主菜按键定义
    void (*Function[SMENU_NO])(void); 子菜单对应的服务程序
    void (*FunctionEx)(void); 菜单回调函数
} _GuiMenuStru;
```

返回值: 0

## 3. uint8 \_MenuEx(\_GuiMenuStru \* menu)

功能说明: 菜单显示

参数说明:

menu 跟上述函数一致,取消回调函数,

返回值:菜单中的按键定义

## 4. uint8 \_List(\_GuiLisStru t \* list)

功能说明: 菜单显示

参数说明:

list:

```
typedef struct
{
    uint32 x;          //开始 X 坐标
    uint32 y;          //开始 Y 坐标
    uint8 isRt;        //是否带窗体
    uint8 with;        //宽度
    uint8 MaxNum;      //一屏幕多少个
    uint8 no;          //总共有多少个
}
```

```

char *title; //标题
char *str[255]; // 主菜单字符串
} _GuiLisStru;返回值: >0 当前选择的序号,=0 就代表取消键

```

## 5. uint8 \_InputBox(\_GuiInputBoxStru \*,inputbox)

功能说明: 窗体输入方式

参数说明:

inputbox:

```

typedef struct
{

```

```

    uint32 top;           位置(左上角的 x 坐标)
    uint32 left;          位置(左上角的 y 坐标)
    uint32 width;         宽度
    uint32 hight;        高度
    uint8 *caption;       标题
    uint8 *context;       内容
    uint8 type;           1 数字输入
    uint8 datelen;        数据长度
    uint8 keyUpDown;      上下键盘是否有效 0 上键=左键, 下键=右键
    uint8 IsClear;        是否输入后清除显示

```

```

} _GuiInputBoxStru;

```

返回值: 键值

```

extern void _SetInputMode(uint32); //

```

```

extern void _DisInputMode(uint32); //输入法是否允许切换

```

## 6. Uint8 \_GetStr(\_GuiInputBoxStru \*)

功能说明: 输入(跟上面函数区别,没有了窗体)

参数说明:

inputbox: 上述结构体

返回值: 键值

## 7. Void \_SetInputMode (uint32)

功能说明: 设置输入方式

参数说明: 1 数字输入, 2 小写字母输入, 3 大写字母输入, 4 汉字输入

返回值: 无

## 8. Void \_DisInputMode (uint32)

功能说明: 是否可以输入时候切换输入法

参数说明: 0, 关闭切换, 1 允许切换

返回值: 无

9. Uint8 \_MessageBoxEx (char \*lpCaption ,char \*str,, uint8 itype)

功能说明：提示框

参数说明：

str;           具体内容

lpCaption;    显示标题

itype          MSG\_OK 表示只有确定, MSG\_OKCANCEL 表示显示确定取消

返回值：IDOK(1), IDCANCEL(2)

10. uint32 ListEx(\_GuiLisStruEx \*)

功能说明：

参数说明：

\_GuiLisEx

返回值：

11. uint8 \_CreateProgressBar(\_ProGressBar \*)

功能说明：创建进度条

参数说明：

\_ProGressBar \*: 指向要创建的进度条的进度条结构体指针

返回值：成功，失败

12. uint8 \_IncreamentProgressBar(\_ProGressBar \*)

功能说明：画进度条

参数说明：

\_ProGressBar \*: 指向要创建的进度条的进度条结构体指针

返回值：成功,失败

13. void \_SetFrontSize(uint8 size)

功能说明：设置点阵

参数说明：

uint8 size:点阵值（常见：12 或 16）

返回值：无

```
typedef struct
{
    WINDOWS *win;     // 所属窗口
    uint32 top;        // 位置(左上角的 x 坐标)
    uint32 left;       // 位置(左上角的 y 坐标)
    uint32 width;      // 宽度
    uint32 hight;      // 高度
    uint32 max;        // 最大值
    uint32 min;        // 最小值
    uint32 value;      // 最小值
    uint8 *caption;    // 定义显示内容 (最大个数受窗口限制)
    uint8 tabindex;
    void (*Function)(void);     // 对应的服务程序
    uint32 stepPt;
```

```

        uint8    step;
    } _ProGressBar;

```

## 八、串口类

### 1. void \_ComSetTran(uint8 Tran)

功能说明：设置通信方式口,必须在其他串口设置的前面。

参数说明：

Tran: 1 串口,2 红外,3 IRDA ,4 RFID

返回值：无

### 2. void \_ComSet(uint8 \*Setting,uint8 comport)

功能说明：串口设置

参数说明：

Setting “115200,N,8,1”

comport: 物理端口 1,机器底部,,2 机器上端(一般是红外口)

返回值：无

### 3. void \_SendComStr(uint8 \*SendStr,uint32 SendSize)

功能说明：数据发送

参数说明：

SendStr 数据内容

SendSize: 数据长度

返回值：无

### 4. uint32 \_GetComStr(uint8\* Getstr,uint32 GetSize,uint32 GetMs)

功能说明：数据接收

参数说明：

Getstr: 数据接收缓冲区

GetSize 要接收的数据长度

GetMs 接收字符的时间,

返回值：接收到的缓冲区的长度

### 5. void \_CloseCom(void)

功能说明：关闭端口

参数说明：无

返回值：无

## 九、按键类

### 1. uint32 \_ReadKey(void)

功能说明：等待读取按键

参数说明：无

返回值：按键值

## 2. uint32 \_GetKeyExt(void)

功能说明：读取按键,该函数只取一次按键值

参数说明：无

返回值：按键值

## 3. void \_SetKey(uint8 ROW\_KEY\_LOG)

功能说明：设置按键值

参数说明：无

返回值：无

注意：按键值修改后需要改回来以下是按键值模板。

```
uint8 ROW_KEY_LOG[3][8]=
{
    {KEY_FUNC, KEY_RIGHT, KEY_ENTER, KEY_3, KEY_6, KEY_9, KEY_DELETE, KEY_NOHOOK},
    {KEY_UP, KEY_NOHOOK, KEY_DOWN, KEY_2, KEY_5, KEY_8, KEY_0, KEY_ONOFF},
    {KEY_KJ, KEY_LEFT, KEY_CANCEL, KEY_1, KEY_4, KEY_7, KEY_DOT, KEY_LIGHT}
};
```

例子：键盘按左键光标左移，键值为 KEY\_LEFT。

当以上模板的 KEY\_LEFT 修改为 KEY\_DELETE，键盘上按左键做删除动作。

# 十、控制类

## 1. void \_OpenRedLight(void)

功能说明：开红灯

参数说明：无

返回值：无

## 2. void \_CloseRedLight(void)

功能说明：关红灯

参数说明：无

返回值：无

## 3. void \_OpenGreenLight(void)

功能说明：开绿灯

参数说明：无

返回值：无



4. void \_CloseGreenLight(void)  
功能说明：关绿灯  
参数说明：无  
返回值：无
5. uint32 \_GetShutDonwTime(void)  
功能说明：获取关机时间,单位秒  
参数说明：无  
返回值：,秒数
6. void \_SetShutDonwTime(uint32 time)  
功能说明：设置关机时间,单位秒  
参数说明：无  
返回值：,无
7. void \_SetBeepOnOff(uint8 onoff)  
功能说明：设置按键声音  
参数说明：  
    Onoff: 1 开声音,0 关声音  
返回值：,无
8. uint32 \_GetMbt(void)  
功能说明：获取主电池电压  
参数说明：无  
返回值：,电压,该电压要除 100
9. uint32 \_GetSbt(void)  
功能说明：获取纽扣电池电压  
参数说明：无  
返回值：,电压,该电压要除 100
10. void \_Beep(void)  
功能说明：发声音  
参数说明：无  
返回值：无
11. void \_OpenLcdBackLight(void)  
功能说明：开背光灯  
参数说明：无  
返回值：无
12. void \_CloseLcdBackLight(void)  
功能说明：关背光灯  
参数说明：无  
返回值：无

13. void \_OpenKeyBackLight(void)

功能说明： 开键盘灯

参数说明： 无

返回值： 无

14. void \_CloseKeyBackLight(void)

功能说明： 关键盘灯

返回值： 无

15. void \_EnableArmIrqInt(void)

功能说明： 打开时钟中断

参数说明： 无

返回值： 无

16. void \_DisableArmIrqInt (void)

功能说明： 关闭时钟中断

参数说明： 无

返回值： 无

17. void OpenFlashLight(void)

功能说明： 打开手电筒

参数说明： 无

返回值： 无

18. void \_CloseFlashLight(void)

功能说明： 关闭手电筒

参数说明： 无

返回值： 无

## 十一、系统菜单类

1. void \_SetBacklightSys (void)

功能说明： 调用系统的液晶对比度

参数说明： 无

返回值： 无

2. void \_SetLCDContrastSys (void)

功能说明： 调用系统的液晶背光亮度

参数说明： 无

返回值： 无

3. void \_ShutDownPower(void)

功能说明： 调用系统的关机函数

参数说明：无

返回值：无

4. void \_CalendarSys(void)

功能说明：调用系统万年历

参数说明：无

返回值：无

5. void \_BatteryDisplaySys(void)

功能说明：调用系统电池显示

参数说明：无

返回值：无

6. void \_CalculatorSys(void)

功能说明：调用系统计算器

参数说明：无

返回值：无

7. void \_SoundSwitchSys(void)

功能说明：调用系统声音管理

参数说明：无

返回值：无

8. void \_FileTransSys(void)

功能说明：调用系统文件通信

参数说明：无

返回值：无

9. void \_FormatSys(void)

功能说明：调用系统菜单中的系统格式化

参数说明：无

返回值：无

10. void \_AutoShutDownMenuSys(void)

功能说明：自动关机菜单

参数说明：无

返回值：无

11. void \_TimeSettingSys(void)

功能说明：系统时间

参数说明：无

返回值：无

12. void \_InfoSys(void)

功能说明：系统信息显示

参数说明：无

返回值：无

13. void \_DispTimeSys(void)

功能说明：主菜单中的时间显示

参数说明：无

返回值：无

14. void \_GetMachineSn(char \*)

功能说明：获取机器编号

参数说明：机器编号缓冲区

返回值：无

15. void \_SetComSever(void)

功能说明：调用系统通讯设置菜单

参数说明：无

返回值：无

16. void \_Reset(void)

功能说明：复位机器

参数说明：无

返回值：无

17. void \_SetFileTran(char flag,char btl)

功能说明：设置默认文件通讯方式

参数说明：

flag:0—2 (USB,COM1,COM2) (0 为 USB , 1 为 COM1, 2 为 COM2)

btl: 0-4(9600,19200,38400,57600,115200) (0 为 9600,1 为 19200,2 为 38400, 3 为 57600,4 为 115200)

返回值：无

18. void \_GetTransCommPort (uint32 \*COM,uint32 \*BTL)

功能说明：获取系统文件通讯端口和速度

参数说明：

COM:0—2 (USB,COM1,COM2) (0 为 USB , 1 为 COM1, 2 为 COM2)

BTL: 0-4(9600,19200,38400,57600,115200) (0 为 9600,1 为 19200,2 为 38400, 3 为 57600,4 为 115200)

返回值：无

19. void \_SetShutDownTimeEx (uint32)

功能说明：设置定时关机，但复位后无效

参数说明：无

返回值：无

20. void \_SoundOn(void)  
功能说明：打开声音  
参数说明：无  
返回值：无
21. void \_SoundOff(void)  
功能说明：关闭声音  
参数说明：无  
返回值：无
22. void RunHex(char \* filename)  
功能说明：运行可执行程序  
参数说明：  
Char\* filename：可执行程序文件名  
返回值：无
23. void \_IsPowerKey(uint8)-  
功能说明：运行二次开发后，关机安检是否生效 0 失效，非 0 生效  
参数说明：  
Uint8: 接收缓冲区;  
返回值：无

## 十二、RFID 类(无线射频识别)

1. void \_RfidInit(void);  
功能说明：打开电源并初始化模块  
参数说明：无  
返回值：返回卡类型
2. uint16 \_RfidFind(uint8 active) (uint8 cardid[])  
功能说明：寻卡并返回卡类型  
参数说明：无  
返回值：返回卡类型
3. uint16 \_RfidReadCardId(uint8 cardid[])  
功能说明：读卡片序列号  
参数说明：  
cardid :卡片序列号存放地址 长度大于 4  
返回值：返回卡类型
4. uint16 \_RfidSelectCard(uint8 cardid[]))  
功能说明：选定卡片  
参数说明：  
cardid 卡号 4 字节

返回值:

正确: 卡片容量

错误: -1

5. uint16 \_RfidCheckPwd(uint8 pwd[],uint8 cardid[],uint8 address,uint8 type))

功能说明: 验证卡片密码

参数说明:

pwd:密码 6 个字节

cardid : 卡号 4 个字节

address: 块地址

type: 密码类型

返回值:

0x00 = OK

0x02 = 错误

6. uint16 \_RfidWriteData(uint8 data[],uint8 address)

功能说明: 写数据到 M1 卡某一块

参数说明:

data 16 字节数据

address 块地址

返回值:

0x00 = OK

0x02 = 错误

7. uint16 \_RfidReadData(uint8 data[],uint8 address)

功能说明: 读 M1 卡某一块的数据

参数说明:

data 16 字节数据存放指针

address 块地址

返回值:

0x00 = OK

0x02 = 错误

8. uint16 \_RfidCharge(uint32 value,uint8 address,uint8 type)

功能说明: 扣款和充值

参数说明:

data 16 字节数据存放指针

address 块地址

type 扣款或者充值

返回值:

0x00 = OK

0x02 = 错误

9. uint16 \_RfidBackup(uint8 address)

功能说明：备份钱包

参数说明：

address 块号

返回值：

0x00 = OK

0x02 = 错误

#### 10. uint16 \_RfidDormancy(void)

功能说明：命令卡片进入休眠状态

参数说明：无

返回值：

0x00 = OK

0x02 = 错误

#### 11. uint16 \_RfidReset(void)

功能说明：复位 RFID 模块

参数说明：无

返回值：

0x00 = OK

0x02 = 错误

#### 12. uint16 \_RfidOpenAntenna(void)

功能说明：开启天线

参数说明：无

返回值：返回卡类型

#### 13. uint16 \_RfidCloseAntenna(void)

功能说明：关闭天线

参数说明：无

返回值：返回卡类型

#### 14. void \_RfidClose(void)

功能说明：关闭天线

参数说明：无

返回值：返回卡类型

#### 15. Int16 \_RfidInitWallet(uint32 value,uint8 address)

功能说明：初始化钱包

参数说明：

uint32 value: 初始化的值

uint8 address: 块地址

返回值：成功 0，失败 非 0

#### 16. uint32 \_RfidReadWallet(uint8 address)

功能说明：读钱包

参数说明：

uint8 address: 块地址

返回值: 读取到的值

17. int \_Rfid125Read(uint8\* id, int32 len, int32 ms)

功能说明: 读 125 模块

参数说明: id: 方向 (out) 存放卡号的地址 长度 >= len

len: 预读卡号的长度

ms: 超时时间

返回值: 实际读取到的长度

18. int \_Rfid125Start(void)

功能说明: 打开 125 模块

参数说明: 无

返回值: 成功 1, 失败 0

19. void \_Rfid125End(void)

功能说明: 关闭 125 模块

参数说明: 无

返回值: 无

### 十三、扫描条形码类

1. Unit8 \_ScanBarcodeKey(unit8 \*barcode, unit8 evkey)

功能说明: 按着 evkey 键出现红光, 放开 evkey 键红光消失

参数说明:

Barcode 读入的条形码赋值给 Barcode

Evkey 要读入条形码需要按的键

返回值: 无

2. Uint8 \_ScanBarcode(unit8 \*barcode)

功能说明: 调用该函数出现红光, 当得到条码值后马上消失, 或者没有得到值时 30 秒后消失

参数说明:

Barcode 读入的条形码赋值给 Barcode

返回值: 无

### 十四、蓝牙函数 (A380 机器)

1. Unit8 \_BtOpen(void)

功能说明: 打开蓝牙设备

参数说明: 无

返回值:

0 ---成功

0 ---失败

2. Void \_BTClose(void)

功能说明: 关闭蓝牙设备



参数说明：无

返回值：无

### 3. Void \_BtSetMode(unit8 atmdl)

功能说明：设置蓝牙工作模式

参数说明：

Atmdl: 0 --AT 指令模式

1 --数据模式

返回值：无

### 4. Unit8 \_BtGetModel(void)

功能说明：获取蓝牙设备工作模式

参数说明：无

返回值：

### Int8 BtScan(unit8 num, char \*cClass, char \*cIac)

功能说明：扫描设备

参数说明：

num: 最多蓝牙设备响应数

cClass: 蓝牙设备类实际上是一个 32 位的参数，该参数用于指出设备类型，以及所支持的服务类型。为了能有效地对周围诸多蓝牙设备实施过滤，快速查询或被查询自定义蓝牙设备，用户可以将模组设置为非标准蓝牙设备类，

如：0x1f1f（十六进制）默认值：0

cIac: 查询访问码实际上是一个 32 位的参数。默认值：9e8b33

返回值：搜索到的设备数

### 5. Int8 BtName(char \*name, unit8 op)

功能说明：设置、获取本地蓝牙设备的名称

参数说明：

Name: 要设置的名称或者获取到的名称

Op: 0-获取；1-设置

返回值：

0-成功

<0 错误

>0 获取的名称长度

### 6. int8 \_BtRName(char \*name, char \*addr)

功能说明：获取远程蓝牙设备的名称

参数说明：

Name: 获取远程蓝牙设备的名称

Addr: 远程蓝牙地址

返回值：获取到的名称长度

## 7. int8 \_BtAddr(char \*addr)

功能说明：获取本地蓝牙设备的地址

参数说明：

addr: 获取的地址

返回值：地址长度

## 8. int8 \_BtPair(char \*addr,uint16 ms)

功能说明：蓝牙设备配对

参数说明：

Addr: 远程设备蓝牙地址

Ms: 配对超时（秒）

返回值：

0 --成功

<0 --错误

## 9. int8 \_BtLink(char \*addr)

功能说明：连接蓝牙设备

参数说明：

Addr: 远程设备蓝牙地址

返回值：

0 --成功

<0 --错误

## 10. int8 \_BtPswd(char \*pswd,uint8 op)

功能说明：设置/查询配对码（密码）

参数说明：

pswd: 设置或者查询到的配对码

op: 0-获取；1-设置

返回值：

0- 成功

<0 错误

>0 获取的名称长度

## 11. uint32 \_BtGetData(uint8 \*RecBuff,uint32 len,uint16 ms)

功能说明：获取蓝牙数据

参数说明：

RecBuff: 数据存放地址

Len: 要获取的数据长度

Ms: 超时时间

返回值：

实际获取的数据长度

## 12. int8 \_BtDefault(void)

功能说明：恢复蓝牙模块默认设置

① 设备类： 0

② 查询码： 0x009e8b33

③ 模组工作角色： Slave Mode（从动模式）

④ 连接模式：任意蓝牙设备连接模式

⑤ 串口参数：波特率—38400bits/s;停止位：1 位；校验位：无

⑥ 配对码：“1234”

⑦ 设备名称：HJ

⑧ 设备过滤准则：0x00

参数说明：无

返回值：

0 --成功

<0 --错误

## 13. int16 \_BtSendData(uint8 \* Buff,uint32 Len)

功能说明：发送数据

参数说明：

Buff: 数据地址

Len: 要发送的数据长度

返回值：

返回实际发送的长度

-1 错误

## 14. int8 \_BtSetRole(uint8 role)

功能说明：设置模块角色

参数说明：

role:

0 从模式

1 主模式

返回值:

正确 0

错误 -1

15. int8 \_BtMrad(char \*addr)

功能说明: 获取最近使用过的蓝牙地址

参数说明:

char \* addr: 存储蓝牙地址的缓冲区

返回值: 成功 0, 失败 非 0

16. int8 \_BtFsad(char \*addr)

功能说明: 从蓝牙配对列表中查找指定的认证设备

参数说明:

char \* addr: 指定的认证设备地址

返回值: 成功 0, 失败 非 0

17. void \_BtGetList(void)

功能说明: 获取蓝牙设备列表

参数说明: 无

返回值: 无

## 十五、WIFI 函数(A380 机器)

1. int8 \_WifiAbrd(void)

功能说明: WIFI 模块自动匹配波特率

参数说明: 无

返回值:

正确: 0

错误: -1

2. int8 \_WifiUpgrade(void)

功能说明: 跳过 Firmware Upgrade 操作

参数说明: 无

返回值:

正确: 0

错误: -1

3. int8 \_WifiLoadDone (void)

功能说明：Load booting 操作  
参数说明：无  
返回值：  
    正确： 0  
    错误： -1

4. int8 \_WifiBand(uint8 band\_val)

功能说明：设置 WIFI 模块工作频率 2.4G/5G  
参数说明：  
    band\_val:  
        =0 : 2.4G  
        =1 : 5G  
返回值：  
    正确： 0  
    错误： -1

5. int8 \_WifiInit(void)

功能说明：初始化 WIFI 模块，在\_WifiBand 后执行  
参数说明：无  
返回值：  
    正确： 0  
    错误： -1

6.int8 \_WifiPassscan(uint16 bit\_map)

功能说明：配置 WIFI 模块扫描的频道  
参数说明：

bit\_map:

Channel[n]	Channel[n-1]	.....	Channel[4]	Channel[3]	Channel[2]	Channel[1]
0	0	.....	0	0	0	0

相应位为 0 时，允许扫描，相应位为 1 时禁止扫描

返回值：  
    正确： 0  
    错误： -1

## 7. int8 \_WifiSetNetworkType(char \*Type)

功能说明：设置 WIFI 模块工作模式

参数说明：

Type:

“INFRASTRUCTURE” -基础模式

“IBSS,type,chan\_num” -Ad-Hoc 模式，安全关

“IBSS\_SEC,type,chan\_num3” -Ad-Hoc 模式，安全开支持 WEP

返回值：

正确：0

错误：-1

## 8. int8 \_WifiPreSharedKey(char \*psk)

功能说明：设置 WIFI 密码

参数说明：

psk: 密码

返回值：

正确：0

错误：-1

## 9. int8 \_WifiScan(uint8 chan\_num,char \*ssid, Access\_Points \*ap,uint8

ap\_num)

功能说明：扫描 WIFI 热点

参数说明：

chan\_num: 扫描的频段，为 0 时，扫描所有频段

ssid: 扫描指定的 SSID

ap: 扫描结果返回的地址（OUT）

ap\_num: ap 中可容 SSID 的最大数量

typedef struct

```
{  
    char uSSID[32];  
    uint8 uMode;  
    uint8 uRSSI;  
}Access_Points;
```

返回值：

扫描到的 AP 数量

## 10.int8 \_WifiAuthmode(uint8 value)

功能说明：配置认证模式

参数说明：

Value:

0 - Open Key Authentication (认证)

1 – Shared Key Authentication

返回值：

正确：0

错误：-1

## 11. int8 \_WifiJoin(char \*ssid\_name,uint8 TxRate,uint8 TxPower)

功能说明：加入 ssid\_name 网络

参数说明：

ssid\_name,: 热点 SSID 的名称

TxRate: 连接带宽

Data Rate (Mbps)	Value of uTxDataRate
Auto-rate	0
1	1
2	2
5.5	3
11	4
6	5
9	6
12	7
18	8
24	9
36	10
48	11
54	12
MCS0	13
MCS1	14
MCS2	15
MCS3	16
MCS4	17
MCS5	18
MCS6	19
MCS7	20

TxPower: 天线输出功率

- 0 – Low power (7dBm)
- 1 – Medium power (10dBm)
- 2 – High power (16 to 17dBm)

返回值:

- 正确: 0
- 错误: -1

## 12. int8 \_WifiSetIp(uint8 dhcp,char \*ip,char \*subnet,char \*gateway)

功能说明: 设置 WIFI 模块的 IP

参数说明:

dhcp: IP 获取模式

- 0 – Manual
- 1 – DHCP
- 2 – Auto-IP

ip: IP 地址

subnet: 子网掩码

gateway: 网关

返回值:

- 正确: 0
- 错误: -1

## 13. int8 \_WifiCreateSocket(int8 type ,char \*ipaddr,char \*portno,char \*lport)

功能说明: 建立 Socket 连接

参数说明:

type:

- 0: TCP (建立 TCP 连接)
- 1: UDP (建立 UDP 连接)
- 2: LTCP (建立 TCP 监听)
- 3: LUDP (建立 UDP 监听)

ipaddr: 远程 IP 地址

portno: 远程端口

lport: 本地监听端口

返回值:

- 正确: 建立的 Socket 连接号
- 错误: 0



#### 14. int8 \_WifiCloseSocket(int8 socketno)

功能说明：关闭 Socket 连接

参数说明：

Socketno: 已建立的 Socke 连接号

返回值：

正确：0

错误：-1

#### 15. int16 \_WifiSendData(int8 sockno,uint16 slength,char \*sdata,char \*dip,char \*dport)

功能说明：发送数据

参数说明：

sockno: 已建立的 Socke 连接号

slength: 发送的数据长度

sdata: 发送的数据

dip: 目标 IP <TCP/UDP 为"0">

dport: 目标端口<TCP/UDP 为"0">

返回值：

正确：发送的数据长度

错误：<0

#### 16. int16 \_WifiReceiveData(int8 \*sockno,char \*sdata,uint16 ms)

功能说明：接收数据

参数说明：

sockno: 已建立的 Socke 连接号

sdata: 接收的数据地址

ms: 接收超时时间

返回值：

接收的数据长度

#### 17. void \_WifiClose(void)

功能说明：关闭 WIFI 模块

参数说明：无

返回值：无

#### 18. int8 \_WifiOpen(void)

功能说明：打开 WIFI 模块

参数说明： 无

返回值： 成功 1，失败 0

## 十六、GPRS 函数（A380 机器）

### 1. int16 \_GprsGetSignal(char \*cResult)

功能说明：信号质量查询

参数说明：cResult 字符串返回查询结果(out,即时将查询结果存储在 cResult 指针所指向的字符串中)，如不需要可设置 NULL

返回值：

0 小于等于-115dBm

1-111dBm

2...30-110...-54dBm

31 大于等于-52dBm

99 未知或者不可测

其他则错误

### 2. int16 \_GprsGetRegistration(char \*cResult)

功能说明：网络注册信息查询

参数说明：cResult 字符串返回查询结果(out)，如不需要可设置 NULL

返回值：

0 没有注册，MT 当前没有寻找一个运营商去注册

1 注册的，归属网络

2 未注册，但是 MT 在寻找一个新的运营商去注册

3 注册被拒

4 未知

5 注册的，漫游中

其他则错误

### 3. int16 \_GprsGetCurrentSelection(char \*cResult)

功能说明：获取当前运营商

参数说明：cResult 字符串返回查询到运营商名称(out)

返回值：

0 自动模式

1 手动模式

4 手动/自动

其他则错误

### 4. int16 \_GprsGetAllSelection(char \*cResult)

功能说明：获取网络中存在的运营商

参数说明：cResult 字符串返回查询到运营商名称(out)

返回值：

正确：>=0

错误: <0

5. int16 \_GprsGetSimPinStatus(char \*cResult)

功能说明: 查询是否设置 PIN 码

参数说明: cResult 字符串返回查询结果(out) , 如不需要可设置 NULL

返回值:

- 0 -READY            MT 不需要任何密码
- 1 -SIM PIN         MT 在等待输入 SIM 卡的 PIN 码
- 2 -SIM PUK    MT 在等待输入 SIM 卡的 PUK 码
- 3 -PH\_SIM PIN    ME 等待提供从手机到 SIM 卡的密码(防盗)
- 4 -PH\_SIM PUK    ME 在等待 SIM 的 PUK 码的输入(防盗)
- 5 -SIM PIN2
- 6 -SIM PUK2

其他则错误

6. int16 \_GprsSetSimPin(char \*cPin,char \*cNewPin)

功能说明: 设置 PIN 码

参数说明:

- cPin 旧的 PIN 码(in)
- cNewPin 新的 PIN 码(in)

返回值:

- 正确: >=0
- 错误: <0

7. int16 \_GprsSetService(uint8 uOperate)

功能说明: 设置 GPRS 附着或者分离

参数说明:

- uOperate:
- 0 分离
- 1 附着

返回值:

- 正确: >=0
- 错误: <0

8. int16 \_GprsGetService(char \*cResult)

功能说明: 获取 GPRS 附着状态

参数说明: cResult 字符串返回查询结果(out) , 如不需要可设置 NULL

返回值:

- 0 分离
- 1 附着
- 其他则错误

9. int16 \_GprsSetMultiIp(uint8 uOperate)

功能说明: 设置 IP 连接链路模式

参数说明:

uOperate :

0 , 单个连接

1 , 多路连接, 最大支持 7 个连接

返回值:

正确:  $\geq 0$

错误:  $< 0$

#### 10. int16 \_GprsGetMultiIp(void)

功能说明: 获取当前链路模式

参数说明: 无

返回值:

0, 单个连接

1, 多路连接

其他则错误

#### 11. int16 \_GprsStartTask(char \*cApn,char \*cName,char \*cPassword)

功能说明: 启动任务并设置接入点 APN、用户名、密码

执行后,状态变为 IP START。

参数说明:

cApn: 接入点

“CMWAP”/“CMNET” 中国移动

“UNIWAP”/“UNINET” 中国联通

cName: 用户名

cPassword: 密码

返回值:

正确:  $\geq 0$

错误:  $< 0$

#### 12. int16 \_GprsActivatePDP(void)

功能说明: 激活移动场景, 发起 GPRS 或 CSD 无线连接

只有在 IP START 状态下才能激活移动场景, 执行后状态变为 IP CONFIG。

在模块接受激活场景操作后, 如果移动场景激活成功, 状态变为 IP GPRSACT, 返回 OK, 否则返回 ERROR。

参数说明: 无

返回值:

正确:  $\geq 0$

错误:  $< 0$

#### 13. int16 \_GprsGetLocalIp(char \*ip)

功能说明: 获取本地 IP 地址

参数说明: ip 字符串返回 IP(out) , 如不需要可设置 NULL

返回值:

正确:  $\geq 0$

错误:  $< 0$

#### 14. int16 \_GprsSetConnection(char\*cMode,char \*cIp, char \* cPort,uint16 ms)

功能说明: 建立 TCP 连接或注册 UDP 端口号

参数说明:

CMode:

TCP 建立 TCP 连接

UDP 建立 UDP 连接

cIp: 远端服务器 IP 地址

cPort: 远端服务端口

ms: 超时时间

返回值:

0 IP\_INITIAL

1 IP\_START

2 IP\_CONFIG

3 IP\_GPRSACT

4 IP\_STATUS

5 TCP\_CONNECTING

6 CONNECT\_OK

7 TCP\_CLOSING

8 TCP\_CLOSED

9 PDP\_DEACT

其他则错误

#### 15. int16 \_GprsGetIpStatus(int8 iMultiIp,char \*cStatus)

功能说明: 获取当前 IP 状态

参数说明:

iMultiIp:

0-7: 多路链接

-1: 单路链接

cStatus: 字符串返回查询结果(out) , 如不需要可设置 NULL

返回值:

0 IP\_INITIAL 初始化

1 IP\_START 启动任务

2 IP\_CONFIG 配置场景

4 IP\_STATUS 获得本地 IP 地址

5 TCP\_CONNECTING/UDP\_CONNECTING/SERVER LISTENING TCP 连接中/UDP

连接中/服务器监听中

6 CONNECT OK 连接建立成功

7 TCP\_CLOSING/UDP\_CLOSING 正在关闭 TCP 连接/正在注销 UDP 端口

8 TCP\_CLOSED/UDP\_CLOSED TCP 连接断开/UDP 端口注销

9 PDP\_DEACT 场景被释放

其他则错误

16. int16 \_GprsSendData(uint8 \*cData,uint16 iLength)

功能说明：单链路 发送数据（Non-transparent mode）

参数说明：cData: 数据地址 (in)

iLength: 数据长度

返回值：

正确：>=0

错误：<0

17. int16 \_GprsIpClose(void)

功能说明：关闭 TCP/UDP 连接

参数说明：无

返回值：

正确：>=0

错误：<0

18. int16 \_GprsDeactivatePDP(void)

功能说明：关闭场景

参数说明：无

返回值：

正确：>=0

错误：<0

19. int16 \_GprsSetTcpipMode(uint8 uMode)

功能说明：设置 TCPIP 透明非透明模式

参数说明：

uMode :

0 非透明模式

1 透明模式

返回值：

正确：>=0

错误：<0

20. int16 \_GprsGetTcpipMode(void)

功能说明：获取 TCPIP 透明非透明模式

参数说明：无

返回值：

0 非透明模式

1 透明模式

其他则错误

21. int16 \_GprsGetAtStatus(void)

功能说明：获取模块工作状态

参数说明：无

返回值：

0 AT

1 DATA

其他则错误

## 22. int16 \_GprsSetDefault(void)

功能说明：恢复默认参数

参数说明：无

返回值：

正确：>=0

错误：<0

## 23. int8 \_GprsOpen(uint32 iBaud)

功能说明：打开模块

参数说明：

iBaud: 波特率

返回值：

正确：>=0

错误：<0

## 24. int8 \_GprsClose(void)

功能说明：关闭模块

参数说明：无

返回值：

正确：>=0

错误：<0

## 25. int16 \_GprsSetSingleConnection(char \*cApn,char\*cMode,char \*cIp, char\* iPort)

功能说明：建立一个单连接的 TCP/UDP 连接

参数说明：

cApn: 接入点名

cMode:

TCP 建立 TCP 连接

UDP 建立 UDP 连接

cIp: 远端服务器 IP 地址

iPort: 远端服务端口

返回值：

正确：= 0

错误：<0

## 十七、其他类

## 1. void \_DoubleToStr(char \*s1 , double s2 ,int n )

功能说明：类型转换

参数说明：

s1: 字符缓冲区

s2: 要转换的 DOUBEL

n: 小数点位数

返回值：指向 S1 的指针

## 2. void \_GetSysVer(char \* ver,char \*UpdateTime)

功能说明：获取系统版本更新日期

参数说明：

ver 版本号

UpdateTime 更新时间

返回值：无

## 十八：C 语言函数标准函数(参考 C 语言手册)

(提示：在您的开发中，请使用以下函数替换相应标准 C 函数)

推荐函数	可替换的标准 C 函数
_strcat	strcat
_strncat	strncat
_ memcmp	memcmp
_ strcmp	strcmp
_strncmp	strncmp
_strxfrm	strxfrm
_ strchr	strchr
_ strcspn	strcspn
_ strpbrk	strpbrk
_ strchr	strchr
_ strstr	strstr
_memset	memset
_ strlen	strlen
_ strcpy	strcpy
_strncpy	strncpy
_atof	atof
_abs	abs
_free	free



_malloc	malloc
_realloc	realloc
_memcpy	Memcpy
_strlen	strlen
_sprintfEx	sprintfEx
_sprintfxy	sprintfxy

## 十九、与电脑接口动态库说明

### 数据下装

函数名称: **SendFileToHc**

传入参数

顺序	代码	类型	说明
1	PORT	char	通讯端口号, 参见标准代码 PORT_NO
2	BAUDRATE	char	通讯波特率, 参见标准代码 BAUD_RATE
3	DATAFILE	char	需要下装到抄表机中的文件名称(带路径)。如果存在多个需要下装的文件, 则用“;”进行分隔。
4	APPFILE	Char	需要下装到抄表机的机内程序文件名称(带路径)。如果存在多个需要下装的文件, 则用“;”进行分隔。
5	PROMPT	char	1

**n 返回参数**

返回参数数据类型为 int, 各返回值代码的意义如下:

代码	说明
0	成功。
1	搜寻端口出错。
2	指定的端口不能打开。
3	超时
4	文件传输出现错误
999	其他

**n 示例**

int **SendFileToHc**(char **PORT**,char **BAUDRATE**,char **DATAFILE**, char **APPFILE**, char

**PROMPT);**

**SendFileToHc('1', '05', 'C:\F1.DBF; F2.DBF', 'C:\F3.dwn; F4.mid', '1');**

注：参数 1：'1'表述端口号为 *COM1*，见标准代码 *PORT\_NO*

参数 2：'05'表示波特率为 *115.2k*，见标准代码 *BAUD\_RATE*

参数 3：F2.DBF 的路径同 F1.DBF 的路径。

参数 4：F4.mid 的路径同 F3.dwn 的路径。

## 数据上装

**n 函数名称：SendFileToPc**

**n 传入参数**

顺序	代码	类型	说明
1	PORT	char	通讯端口号，参见标准代码 <i>PORT_NO</i>
2	BAUDRATE	char	通讯波特率，参见标准代码 <i>BAUD_RATE</i>
3	DATAFILE	char	需要从抄表机中取的文件名，如果存在多个文件，则用 “;” 进行分隔。
4	FILESAVEAS	char	另存为路径与文件名。如果 <i>FILENAME</i> 中存在多个文件，则 <i>FILESAVEAS</i> 中的文件名与之一一对应，并用 “;” 进行分隔。
5	PROMPT	char	1

**n 返回参数**

返回参数数据类型为 *int*，各返回参数代码的意义如下：

代码	说明
0	成功。
1	搜寻端口出错。
2	指定的端口不能打开。
3	超时
4	文件传输出现错误
999	其他

**n 示例**

```
int SendFileToPc(char PORT,char BAUDRATE,char DATAFILE, char FILESAVEAS,
char PROMPT);
```

```
SendFileToPc('1','05','F1.DBF;F2.DBF','C:\F1.DBF; F2.DBF','1');
```

注：参数1：'1'表述端口号为COM1，见标准代码PORT\_NO

参数2：'05'表示波特率为115.2k，见标准代码BAUD\_RATE

参数3：F2.DBF 的路径同 F1.DBF 的路径。

### 获取抄表机序列号

**n 函数名称：GetHcNum**

**n 传入参数**

顺序	代码	类型	说明
1	PORT	char	通讯端口号，参见标准代码 PORT_NO
2	BAUDRATE	char	通讯波特率，参见标准代码 BAUD_RATE
3	HCNUM	char	存放抄表机序列号返回值，传入给空。 相同型号的抄表机，抄表机序列号不允许重复

**n 返回参数**

返回参数数据类型为 int，各返回参数代码的意义如下：

代码	说明
0	成功。
1	搜寻端口出错。
2	指定的端口不能打开。
3	超时
4	文件传输出现错误
999	其他

**n 示例**

```
int GetHcNum(char PORT,char BAUDRATE,char HCNUM);
```

```
GetHcNum('1','05',HCNUM);
```

注：参数1：'1'表述端口号为COM1，见标准代码PORT\_NO

参数2：'05'表示波特率为115.2k，见标准代码BAUD\_RATE

## 设置抄表机系统时间

**n** 函数名称: **SetDateTime**

**n** 传入参数:

顺序	代码	类型	说明
1	PORT	char	通讯端口号, 参见标准代码 PORT_NO
2	BAUDRATE	char	通讯波特率, 参见标准代码 BAUD_RATE
3	DATE	char	格式: YYYY-MM-DD
4	TIME	char	格式: HH24: MI: SS

**n** 返回参数

返回参数的数据类型为 int, 各返回参数代码的意义如下:

代码	说明
0	成功。
1	搜寻端口出错。
2	指定的端口不能打开。
3	超时
4	文件传输出现错误
999	其他

**n** 示例

```
int SetDateTime(char PORT,char BAUDRATE,char DATE,char TIME);
```

```
SetDateTime('1','05','2008-01-01','08:18:18');
```

注: 参数 1: '1' 表述端口号为 COM1, 见标准代码 PORT\_NO

参数 2: '05' 表示波特率为 115.2k, 见标准代码 BAUD\_RATE

## 获取机器里面目录

**n** 函数名称: *GetHcDir*

**n** 传入参数:

顺序	代码	类型	说明
1	PORT	char	通讯端口号, 参见标准代码 PORT_NO
2	BAUDRATE	char	通讯波特率, 参见标准代码 BAUD_RATE

3	<i>PATH</i>	char	路径根目录为/, 根目录下面例如/test
4	<i>dir</i>	Char*	文件目录

**typedef struct**

**{**

**char cName[256];** 文件名，目录名

**char cMtime[20];** 修改时间

**char cCtime[20];** 创建时间

**char cSize[20];** 文件大小

**int nType;**//F 文件，其他目录

**DWORD iSize;** 文件大小

**DWORD iMtime;** 修改时间

**DWORD iCtime;**创建时间

**}FILE\_DIR**

**n** 返回参数 成功就是文件数量

## 建目录

**n** 函数名称: *CreateHcDir*

**n** *CreateHcDir(char \*PORT,char \*BAUDRATE,char \*PDADIR)ir*

**n** 传入参数:

顺序	代码	类型	说明
1	PORT	char	通讯端口号，参见标准代码 PORT_NO
2	BAUDRATE	char	通讯波特率,参见标准代码 BAUD_RATE
3	<i>PDADIR</i>	char	目录TEST1;TEST2，建立2个目录

**n** 返回参数 成功返回 0

## 删除目录

**n** 函数名称: *DelHcDir*

**n** *DelHcDir (char \*PORT,char \*BAUDRATE,char \*PDADIR)*

**n** 传入参数:

顺序	代码	类型	说明
1	PORT	char	通讯端口号, 参见标准代码 PORT_NO
2	BAUDRATE	char	通讯波特率, 参见标准代码 BAUD_RATE
3	<i>PDADIR</i>	char	目录TEST1;TEST2, 删除2个目录

**n** 返回参数 成功返回 0

## 删除文件

**n** 函数名称: *DelHcFile*

**n** *DelHcFile (char \*PORT,char \*BAUDRATE,char \*PDADIR)*

**n** 传入参数:

顺序	代码	类型	说明
1	PORT	char	通讯端口号, 参见标准代码 PORT_NO
2	BAUDRATE	char	通讯波特率, 参见标准代码 BAUD_RATE
3	<i>PDADIR</i>	char	如 /test/tzh.txt

**n** 返回参数 成功返回 0

### 参数说明

PORT\_NO    0    USB  
PORT\_NO    1    COM1  
PORT\_NO    2    COM2  
PORT\_NO    3    COM3  
PORT\_NO    4    COM4  
BAUD\_RATE 01   9600

BAUD\_RATE 02 19200  
BAUD\_RATE 03 38.4k  
BAUD\_RATE 04 57.6k  
BAUD\_RATE 05 115.2k

## n 宏定义

顺序	宏名	字符串	说明
1	KEY_0	‘0’	0 键
2	KEY_1	‘1’	1 键
3	KEY_2	‘2’	2 键
4	KEY_3	‘3’	3 键
5	KEY_4	‘4’	4 键
6	KEY_5	‘5’	5 键
7	KEY_6	‘6’	6 键
8	KEY_7	‘7’	7 键
9	KEY_8	‘8’	8 键
10	KEY_9	‘9’	9 键
11	KEY_DOT	‘.’	. 键
12	KEY_KJ	‘*’	快捷键
13	KEY_UP	‘U’	向上导航
14	KEY_DOWN	‘D’	向下导航
15	KEY_LEFT	‘L’	左向导航
16	KEY_RIGHT	‘R’	右向导航
17	KEY_CANCEL	‘C’	退出键
18	KEY_DELETE	‘T’	删除键
19	KEY_ENTER	‘E’	确认键
20	KEY_LIGHT	‘G’	背光灯
21	KEY_FUNC	‘F’	功能键
22	KEY_ONOFF	‘O’	开关机按键
23	KEY_NOHOOK	0xFF	没有按键
24	KEY_SCAN	‘S’	扫描键
25	KEY_TAB	‘B’	换挡键

26	S_IFMT	0170000	
27	S_IFDIR	0040000	
28	S_IFREG	0100000	
29	MMENU_NO	8	
30	SMENU_NO	8	
31	MSG_OK	0	
32	MSG_OKCANCEL	1	
33	IDOK	1	
34	IDCANCEL	2	
35	DIDOK	0	
36	DCANCEL	2	
37	RFID_TIMEOUT	0xFF	无
38	RFID_ERROR	0xFF	无
39	RFID_OK	0x00	无
40	RFID_ALL_CARD	0x52	无

## 板上接口

（函数设置请看前面串口函数）



