

短信动态库说明

1 对外接口函数说明

一、启动服务 `int _stdcall SMSStartService(int nPort,DWORD BaudRate = 57600, int Parity=2, int DataBits = 8,int StopBits=0,int FlowControl=0,char* csca="card")`

参数: nPort 串口号 如 1 则表示 COM1
BaudRate 波特率 115200
Parity 校验位 2
DataBits 数据位 8
StopBits 停止位 0
FlowControl 流控制 0
Csca 短信中心号码, 可以使用默认值,若设置则格式如: "+8613800591500"

返回值: 非 0 成功, 0 失败

二、发送短消息 `DWORD _stdcall SMSSendMessage(char* Msg,char* PhoneNo)`

参数: Msg 消息内容,如果为中文则一条最多 70 个字, 多于 70 个字分多条短信发送
如果全为英文则一条最多为 140 个字符, 多于 140 则分多条发送
PhoneNo 目标号码 格式如 "13800591500"

返回有错误时为-1,其它返回 1。

三、接收短消息 `int _stdcall SMSGetNextMessage(SMSMessageStruct* Msg)`

结构体类如下

```
typedef struct _sms_msg_t_
{
    char    cMsg[512];        //短信内容
    char    cPhoneNo[32];    //对方手机号码
    char    cReceTime[32];   //接收时间
}SMSMessageStruct;
```

参数 Msg 读取的短消息

返回 1 有短信 0 无

四、查询发送状态报告 `int _stdcall SMSReport(SMSReportStruct* rept)`

```
typedef struct _sms_report_t_
{
```

```

        DWORD    dwIndex;           //短消息编号:index, 从开始递增
        char      cMsg[512];        //短信内容
        int       iSuccess;         //是否发送成功 0 为失败, 非为成功
        char      cPhoneNo[32];     //目标手机号码
    }SMSReportStruct;

```

参数 rept 短消息状态报告

返回 1 有 0 无

五、停止服务 int _stdcall SMSStopSerice()

六、最近一次错误 int _stdcall SMSGetLastError(char* err)

参数 err 为错误内容

返回错误长度

七、int _stdcall SMSCommand(const char *input, char * output, int sleeptime=300)发送命令。

//参数 input: 要发送的命令字符, output, 设备返回的信息, sleeptime: 等待时间, 即等待 sleeptime 时间并返回设备在 sleeptime 时间内发回的消息。

2 调用方法以及用例

2.1.1 声明

//消息结构体 类型声明

```

typedef struct _sms_msg_t_
{
    char      cMsg[512];           //短信内容
    char      cPhoneNo[32];        //对方手机号码
    char      cReceTime[32];       //接收时间
}SMSMessageStruct;

```

//消息状态报告结构体 类型声明

```

typedef struct _sms_report_t_
{
    DWORD    dwIndex;           //短消息编号:index, 从开始递增
    char      cMsg[512];        //短信内容
    int       iSuccess;         //是否发送成功 0 为失败, 非为成功
    char      cPhoneNo[32];     //目标手机号码
}SMSReportStruct;

```

//启动服务函数类型说明

```

typedef int (_stdcall *pSMSStartServiceFun)(int nPort,DWORD BaudRate = 57600, int
    Parity=2, int DataBits = 8,int StopBits=0,int FlowControl=0,char* csca="card");
//读取短信函数类型说明
typedef int (_stdcall *pSMSGetNextMessageFun)(SMSMessageStruct* Msg);
//发送消息类型说明
typedef DWORD (_stdcall *pSMSSendMessageFun)(char* Msg,char* PhoneNo);
//读取状态报告类型说明
typedef int (_stdcall *pSMSReportFun)(SMSReportStruct* rept);
//停止服务函数类型说明
typedef int (_stdcall *pSMSStopSericeFun)();
//发送命令函数类型说明
typedef int (_stdcall *pSMSCommand)(const char *input, char * output, int sleeptime);

```

2.1.2 调用过程

```

HINSTANCE hDll = LoadLibrary("SMSDLL.dll"); //调用程序目录下须要有此动态库文件
pSMSSendMessageFun SMSSendMessageFun; //函数定义
SMSSendMessageFun = (pSMSSendMessageFun)GetProcAddress(hDll, "SMSSendMessage");
//获取函数指针
If(SMSSendMessageFun != NULL)
    SMSSendMessageFun("短信内容", "手机号码");
//发送短信
FreeLibrary(hDll); //释放动态库

```

3 备注

- 1, SMSStartService 和 SMSStopSerice 应该配对出现，在程序退出之前一定要释放资源
- 2, SMSGetNextMessage 这个需要一个定时器和线程来管理。如果短信多了而不调用此函数，新的短信会覆盖以前的未读取短信