

## SMS DLL Document

### 1 .API

一, **start Service** int \_stdcall SMSStartService(int nPort,DWORD BaudRate = 57600, int Parity=2, int DataBits = 8,int StopBits=0,int FlowControl=0,char\* csca="card")

Param: nPort : COM PORT if 1 equal to COM1

BaudRate: Bandary 115200

Parity: Parity bits 2

DataBits: Data bits 8

StopBits:stop bits 0

FlowControl: Flow Control 0

Csca :SMS Center PhoneNo, Default Format: " +8613800591500"

Return : 1 successful, 0 failed

二, **Send Msg** DWORD \_stdcall SMSSendMessage(char\* Msg,char\* PhoneNo)

Param: Msg:

Msg Content, the Msg contained Chinese length Max. is 70 word.and all English length is 140 character,

PhoneNo:target Phone No, such as "13800591500"

Return :this value will be pointless, you can call function 4 to get successful result or not

三, **Receive Msg** int \_stdcall SMSGetNextMessage(SMSMessageStruct\* Msg)

SMS structure:

```
typedef struct _sms_msg_t_
{
    char Msg[256];           //SMS content
    char PhoneNo[32];        //target phone No
    char ReceTime[32];       //received time
} SMSMessageStruct;
```

Param: Msg

Return: 1 received SMS 0 no received SMS

四, **Get Send-State Report** int \_stdcall SMSReport(SMSReportStruct\* rept)

五, **Stop Service** int \_stdcall SMSStopSerice()

**六、Get Last Error Information**    `int _stdcall SMSGetLastError(char* err)`

Param    `err`: error Information

Return: error Information length

**七、Query SMS Send Success Or Not**    `int _stdcall SMSQuery(DWORD index)`

Param:    `index`:SMS send index

Return: 0 failed;    1 success;    -1 No SMS queried by index,or it still sending .

**八、Send Command**    `int _stdcall SMSCommand(const char *input, char * output, int sleeptime=200)`

Parameters:`input`: command string to send

`output`:information string receive from device

`sleeptime`: waittime.the function will receive the info from device within sleeptime ms.

Return value: return output string length.

## 2. USER NOTES

### 2.1.1 Declaration

//sms struction

`typedef struct _sms_msg_t_`

`{`

`char Msg[256];`            //sms content

`char PhoneNo[32];`    //target phone No

`char ReceTime[32];`        //received time

`} SMSMessageStruct;`

//sms report struction

`typedef struct _sms_report_t_`

`{`

`DWORD index;`            //SMS index, increase from Zero

`char Msg[256];`            //sms Content

`int Success;`            //1 Send SMS successful, other is failed

`char PhoneNo[32];`    //target phone No

`} SMSReportStruct;`

//Declare SMSStartService function

`typedef int (_stdcall *pSMSStartServiceFun)(int nPort,DWORD BaudRate = 57600, int Parity=2,`

```

int DataBits = 8,int StopBits=0,int FlowControl=0,char* csca="card");
// Declare SMSGetNextMessage function
typedef int (_stdcall *pSMSGetNextMessageFun)(SMSMessageStruct* Msg);
// Declare SMSSendMessage function
typedef DWORD (_stdcall *pSMSSendMessageFun)(char* Msg,char* PhoneNo);
// Declare SMSReport function
typedef int (_stdcall *pSMSReportFun)(SMSReportStruct* rept);
// Declare SMSStopSerice function
typedef int (_stdcall *pSMSStopSericeFun)();
//Query whether the SMS has send success by index
Typedef int (_stdcall *pSMSQuery)(DWORD index);

//Send Command string to device
Typedef int (_stdcall *pSMSCommand)(const char *input, char * output, int sleeptime);

```

### 2.1.2 Calling Process

```

HINSTANCE hDll = LoadLibrary("MC8331AT.dll"); //Load Library
pSMSSendMessageFun SMSSendMessageFun; //function define
SMSSendMessageFun = (pSMSSendMessageFun)GetProcAddress(hDll, "SMSSendMessage");

```

```

//get function point
If(SMSSendMessageFun != NULL)
SMSSendMessageFun("Content", "phone");
//Send SMS
FreeLibrary(hDll); //Free Library

```

## 3 Notes

- 1, SMSStartService and SMSStopSerice should match.
- 2, SMSGetNextMessage. Need a thread or a timer to manage