|  |
| --- |
| OpenCPU |
| SDK SMS API说明 |
| 版本1.1日期2025-02-26 |

版权声明

版权所有 © 深圳市有方科技股份有限公司2025。深圳市有方科技股份有限公司保留所有权利。

未经深圳市有方科技股份有限公司书面同意，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

是深圳市有方科技股份有限公司所有商标。

本文档中出现的其他商标，由商标所有者所有。

说明

本文档的使用对象为系统工程师，开发工程师及测试工程师。

本设计指南为客户产品设计提供支持，客户须按照本文中的规范和参数进行产品设计和调试。如因客户操作不当造成的人身伤害和财产损失，有方概不承担责任。

由于产品版本升级或其它原因，本文档内容会在不预先通知的情况下进行必要的更新。

除非另有约定，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

深圳市有方科技股份有限公司为客户提供全方位的技术支持，任何垂询请直接联系您的客户经理或发送邮件至以下邮箱：

Sales@neoway.com

Support@neoway.com

公司网址：http://www.neoway.com

目录

[关于本文档 iv](#_Toc191458221)

[范围 iv](#_Toc191458222)

[读者对象 iv](#_Toc191458223)

[修订记录 iv](#_Toc191458224)

[符号约定 iv](#_Toc191458225)

[1 概述 5](#_Toc191458226)

[2 数据结构 6](#_Toc191458227)

[2.1 typedef enum 6](#_Toc191458228)

[2.2 typedef struct 7](#_Toc191458229)

[3 函数（SMS） 9](#_Toc191458230)

关于本文档

范围

本文档对应产品为N706B系列（RTOS系统）。

读者对象

本文档的使用对象为系统工程师，开发工程师及测试工程师。

修订记录

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 日期 | 变更 | 作者 |
| 1.0 | 2025-01 | 初始版本 | Zhang Yanhong |
| 1.1 | 2025-02 | 修改数据结构章节 | Wang Chen |

符号约定

|  |  |
| --- | --- |
| 符号 | 含义 |
|  | 危险或警告，用户必须遵从的规则，否则会造成模块或客户设备不可逆的故障损坏，甚至可能造成人员身体伤害。 |
|  | 注意，警示用户使用模块时应该特别注意的地方，如不遵从，模块或客户设备可能出现故障。 |
|  | 说明或提示，提供模块使用的意见或建议。 |

# 概述

本文档介绍了OpenCSDK中短消息操作相关的API, 主要包括了以下几个方面.

* 短消息模块的初始化
* 短消息的上报方式及短消息存储位置的获取、设置
* 短信中心号码的获取、设置
* 短消息的读取、删除
* 短消息的发送及接收

# 数据结构

## typedef enum

/\* 短信的三种编码方式，GSM7、8BIT、UCS2 \*/

typedef enum

{

NWY\_SMS\_MSG\_FORMAT\_GSM7 = 1,

NWY\_SMS\_MSG\_FORMAT\_8BIT = 2,

NWY\_SMS\_MSG\_FORMAT\_UCS2 = 3,

}nwy\_sms\_msg\_format\_type\_e;

typedef enum

{

NWY\_SMS\_ENCODING\_NONE = 0,

NWY\_SMS\_ENCODING\_GBK = 1,

} nwy\_sms\_context\_decode\_type\_e;

typedef enum

{

NWY\_SMS\_PP\_IND = 0, // PP msg

NWY\_SMS\_CB\_IND = 1, // CB msg

NWY\_SMS\_CB\_CMAS\_IND = 2,// CB CMAS msg

NWY\_SMS\_SEND\_IND = 3,// CB CMAS msg

}nwy\_mt\_sms\_event\_e;

typedef enum

{

NWY\_SMS\_TRANSFER\_UNKNOWN = -1,

NWY\_SMS\_TRANSFER\_AND\_STORE, //短信上报并存储

NWY\_SMS\_TRANSFER\_ONLY, //短信只上报不存储

} nwy\_sms\_transfer\_type\_e;

typedef enum

{

NWY\_SMS\_TRANSFER\_ONLINE\_UNKNOWN = -1,

NWY\_SMS\_TRANSFER\_ONLINE\_DISCARD = 1,

NWY\_SMS\_TRANSFER\_ONLINE\_CACHE,

NWY\_SMS\_TRANSFER\_ONLINE\_DIRECT,

} nwy\_sms\_transfer\_online\_mode\_e;

typedef enum

{

NWY\_SMS\_STORAGE\_TYPE\_UIM,

NWY\_SMS\_STORAGE\_TYPE\_NV,

NWY\_SMS\_STORAGE\_TYPE\_NONE = -1,

}nwy\_sms\_storage\_type\_e;

typedef enum

{

NWY\_SMS\_MSG\_DFLAG\_NONE = 0,

NWY\_SMS\_MSG\_DFLAG\_READ = 1, //删除已读短信

NWY\_SMS\_MSG\_DFLAG\_READ\_SEND = 2, //删除已读、已发送短信

NWY\_SMS\_MSG\_DFLAG\_READ\_SEND\_UNSEND = 3, //删除已读、未读、已发送短信

NWY\_SMS\_MSG\_DFLAG\_ALL = 4, //删除全部短信

NWY\_SMS\_MSG\_DFLAG\_MAX,

}nwy\_sms\_msg\_dflag\_e;

## typedef struct

typedef struct nwy\_sms\_info\_type{

char phone\_num[NWY\_SMS\_MAX\_ADDR\_LENGTH];

uint32\_t msg\_context\_len;

char msg\_context[NWY\_SMS\_MAX\_MO\_MSG\_LENGTH + 1];

nwy\_sms\_msg\_format\_type\_e msg\_format;

}nwy\_sms\_info\_type\_t;

typedef struct nwy\_sms\_time\_stamp

{

uint16\_t uYear;

uint8\_t uMonth;

uint8\_t uDay;

uint8\_t uHour;

uint8\_t uMinute;

uint8\_t uSecond;

int8\_t iZone;

} nwy\_sms\_time\_stamp\_t;

typedef struct nwy\_sms\_recv\_info\_type{

uint16\_t nIndex;

nwy\_sms\_msg\_format\_type\_e msg\_format; // Message format.

uint32\_t msg\_content\_len;

uint8\_t msg\_pdu[NWY\_SMS\_MAX\_PDU\_LENGTH + 1];

uint8\_t msg\_content[NWY\_SMS\_MAX\_MT\_MSG\_LENGTH + 1];

char source\_phone\_num[NWY\_SMS\_MAX\_ADDR\_LENGTH + 1]; // Source address.

nwy\_sms\_time\_stamp\_t date;

nwy\_sms\_context\_decode\_type\_e context\_decode\_type;

uint8\_t msg\_decoded\_content[NWY\_SMS\_MAX\_MT\_MSG\_LENGTH + 1];

nwy\_sms\_transfer\_type\_e transfer\_type;

nwy\_sms\_storage\_type\_e storage\_type;

nwy\_sim\_id\_e dst\_id;

} nwy\_sms\_recv\_info\_type\_t; /\* Message \*/

typedef struct {

uint8\_t phone\_num[NWY\_SMS\_MAX\_ADDR\_LENGTH];

uint8\_t result;

nwy\_sim\_id\_e dst\_id;

} nwy\_sms\_send\_ind\_t;

typedef struct {

char sca[NWY\_SMS\_MAX\_ADDR\_LENGTH + 1];

unsigned char tosca;

} nwy\_sms\_sca\_t;

typedef struct nwy\_sms\_report\_para{

nwy\_sms\_transfer\_type\_e transfer\_type;

nwy\_sms\_transfer\_online\_mode\_e transfer\_online\_mode;

} nwy\_sms\_report\_para\_t;

typedef struct {

uint8\_t len;

uint8\_t indices[NWY\_MAX\_MSG\_LIST\_LEN];

}nwy\_sms\_msg\_list\_t;

typedef uint32\_t sms\_client\_handle\_t;

typedef struct{

nwy\_sms\_transfer\_type\_e transfer\_type;

nwy\_sms\_recv\_info\_type\_t sms\_info;

}nwy\_sms\_info\_ind\_t;

/\* 短信主动上报注册的回调函数原型 \*/

typedef void (\*nwy\_sms\_evt\_handler)

(

nwy\_sim\_id\_e sim\_id,

nwy\_mt\_sms\_event\_e urc\_type, //事件上报类型

nwy\_sms\_info\_ind\_t \*ind\_struct //事件上报内容

);

# 函数（SMS）

该接口定义位于nwy\_sms\_api.h，用于实现短消息收发以及存储位置设置等功能。

## nwy\_sms\_option\_init

|  |  |
| --- | --- |
| 格式 | nwy\_error\_e nwy\_sms\_option\_init(nwy\_sim\_id\_e sim\_id) |
| 描述 | 初始化短信参数。 |
| 参数说明 | nwy\_sim\_id\_e sim\_id：指定要操作SIM卡的ID |
| 返回值 | 执行结果，成功返回NWY\_SUCCESS，失败返回其他枚举值。 |

## nwy\_sms\_storage\_set

|  |  |
| --- | --- |
| 格式 | nwy\_error\_e nwy\_sms\_storage\_set(nwy\_sim\_id\_e sim\_id, nwy\_sms\_storage\_type\_e sms\_storage) |
| 描述 | 设置短信存储位置。 |
| 参数说明 | nwy\_sim\_id\_e sim\_id：指定要操作SIM卡的ID  nwy\_sms\_storage\_type\_e sms\_storage  1:存储位置为设备侧  2:存储位置为SIM卡 |
| 返回值 | 执行结果，成功返回NWY\_SUCCESS，失败返回其他枚举值。 |

## nwy\_sms\_storage\_get

|  |  |
| --- | --- |
| 格式 | nwy\_error\_e nwy\_sms\_storage\_get(nwy\_sim\_id\_e sim\_id, nwy\_sms\_storage\_type\_e \*sms\_storage) |
| 描述 | 获取短信存储位置。 |
| 参数说明 | nwy\_sim\_id\_e sim\_id：指定要操作SIM卡的ID  nwy\_sms\_storage\_type\_e sms\_storage  1:存储位置为设备侧  2:存储位置为SIM卡 |
| 返回值 | 执行结果，成功返回NWY\_SUCCESS，失败返回其他枚举值。 |

## nwy\_sms\_report\_set

|  |  |
| --- | --- |
| 格式 | nwy\_error\_e nwy\_sms\_report\_set(nwy\_sim\_id\_e sim\_id, nwy\_sms\_report\_para\_t report\_para) |
| 描述 | 设置短信上报方式。 |
| 参数说明 | nwy\_sim\_id\_e sim\_id：指定要操作SIM卡的ID  nwy\_sms\_report\_para\_t report\_para：新短信上报方式相关参数。 |
| 返回值 | 执行结果，成功返回NWY\_SUCCESS，失败返回其他枚举值。 |

## nwy\_sms\_sca\_set

|  |  |
| --- | --- |
| 格式 | nwy\_error\_e nwy\_sms\_sca\_set(nwy\_sim\_id\_e sim\_id, nwy\_sms\_sca\_t sca) |
| 描述 | 设置短信中心号码。 |
| 参数说明 | nwy\_sim\_id\_e sim\_id：指定要操作SIM卡的ID  nwy\_sms\_sca\_t sca：短信中心号码相关信息 |
| 返回值 | 执行结果，成功返回NWY\_SUCCESS，失败返回其他枚举值。 |
| 说明 | 设置前需和当地运营商确认短信中心号码是否正确. |

## nwy\_sms\_sca\_get

|  |  |
| --- | --- |
| 格式 | nwy\_error\_e nwy\_sms\_sca\_get(nwy\_sim\_id\_e sim\_id, nwy\_sms\_sca\_t \*sca) |
| 描述 | 获取短信中心号码。 |
| 参数说明 | nwy\_sim\_id\_e sim\_id：指定要操作SIM卡的ID  nwy\_sms\_sca\_t sca：短信中心号码相关信息 |
| 返回值 | 执行结果，成功返回NWY\_SUCCESS，失败返回其他枚举值。 |

## nwy\_sms\_msg\_send

|  |  |
| --- | --- |
| 格式 | nwy\_error\_e nwy\_sms\_msg\_send(nwy\_sim\_id\_e sim\_id, nwy\_sms\_info\_type\_t \*p\_sms\_data) |
| 描述 | 发送短信。 |
| 参数说明 | nwy\_sim\_id\_e sim\_id：指定要操作SIM卡的ID  nwy\_sms\_info\_type\_t \*p\_sms\_data：发送的短信的相关信息 |
| 返回值 | 执行结果，成功返回NWY\_SUCCESS，失败返回其他枚举值。 |
| 说明 | 返回结果只代表发送短信动作成功,不代表短信在网络上是否发送成功。 |

## nwy\_sms\_recv\_cb\_reg

|  |  |
| --- | --- |
| 格式 | nwy\_error\_e nwy\_sms\_recv\_cb\_reg (nwy\_sim\_id\_e sim\_id, nwy\_sms\_evt\_handler evt\_handler) |
| 描述 | 注册接收短信上报的回调函数 |
| 参数说明 | nwy\_sim\_id\_e sim\_id：指定要操作SIM卡的ID，可以通过sim分别绑定多张卡的短信接收处理函数  nwy\_sms\_evt\_handler evt\_handler：客户需要实现的函数指针，短信内容会通过函数指针的入参传入  typedef void (\*nwy\_sms\_evt\_handler)  (  nwy\_sim\_id\_e sim\_id,  nwy\_mt\_sms\_event\_e urc\_type,  nwy\_sms\_info\_ind\_t \*ind\_struct  ); |
| 返回值 | 执行结果，成功返回NWY\_SUCCESS，失败返回其他枚举值。 |

## nwy\_sms\_recv\_cb\_unreg

|  |  |
| --- | --- |
| 格式 | nwy\_error\_e nwy\_sms\_recv\_cb\_unreg (nwy\_sim\_id\_e sim\_id, nwy\_sms\_evt\_handler evt\_handler) |
| 描述 | 去注册接收短信上报的回调函数 |
| 参数说明 | nwy\_sim\_id\_e sim\_id：指定要操作SIM卡的ID，可以通过sim分别绑定多张卡的短信接收处理函数  nwy\_sms\_evt\_handler evt\_handler：客户需要实现的函数指针，短信内容会通过函数指针的入参传入  typedef void (\*nwy\_sms\_evt\_handler)  (  nwy\_sim\_id\_e sim\_id,  nwy\_mt\_sms\_event\_e urc\_type,  nwy\_sms\_info\_ind\_t \*ind\_struct  ); |
| 返回值 | 执行结果，成功返回NWY\_SUCCESS，失败返回其他枚举值。 |

## nwy\_sms\_msg\_del

|  |  |
| --- | --- |
| 格式 | nwy\_error\_e nwy\_sms\_msg\_del(nwy\_sim\_id\_e sim\_id, uint16\_t index) |
| 描述 | 删除一条短信。 |
| 参数说明 | nwy\_sim\_id\_e sim\_id：指定要操作SIM卡的ID  uint16\_t index:短信索引号 |
| 返回值 | 执行结果，成功返回NWY\_SUCCESS，失败返回其他枚举值。 |

## nwy\_sms\_msg\_read

|  |  |
| --- | --- |
| 格式 | nwy\_error\_e nwy\_sms\_msg\_read(nwy\_sim\_id\_e sim\_id, uint16\_t index, nwy\_sms\_recv\_info\_type\_t \*sms\_data); |
| 描述 | 读取一条短信。 |
| 参数说明 | nwy\_sim\_id\_e sim\_id：指定要操作SIM卡的ID  uint16\_t index:短信索引号  nwy\_sms\_recv\_info\_type\_t \*sms\_data:读取到的短信的内容信息 |
| 返回值 | 执行结果，成功返回NWY\_SUCCESS，失败返回其他枚举值。 |

## nwy\_sms\_msg\_del\_ext

|  |  |
| --- | --- |
| 格式 | nwy\_error\_e nwy\_sms\_msg\_del\_ext(nwy\_sim\_id\_e sim\_id, nwy\_sms\_msg\_dflag\_e delflag) |
| 描述 | 删除一个类型的短信。 |
| 参数说明 | nwy\_sim\_id\_e sim\_id：指定要操作SIM卡的ID  nwy\_sms\_msg\_dflag\_e delflag：删除的短信的类型 |
| 返回值 | 执行结果，成功返回NWY\_SUCCESS，失败返回其他枚举值。 |

## nwy\_sms\_msg\_list\_read

|  |  |
| --- | --- |
| 格式 | nwy\_error\_e nwy\_sms\_msg\_list\_read(nwy\_sim\_id\_e sim\_id, nwy\_sms\_msg\_list\_t \*sms\_list) |
| 描述 | 获取所有短信的下标和类型 |
| 参数说明 | nwy\_sim\_id\_e sim\_id：指定要操作SIM卡的ID  nwy\_sms\_msg\_list\_t \*sms\_list：获取的短信列表 |
| 返回值 | 执行结果，成功返回NWY\_SUCCESS，失败返回其他枚举值。 |