

UDS 接口使用手册

CAN 接口卡系列产品

TN01010101

使用手册

类别	内容
关键词	CAN、CANFD、UDS
摘要	本文档用于说明 UDS 函数库的接口、使用方法以及注意点。



修订历史

版本	日期	原因
V1.00	2020/07/14	创建文档

目录

1. 基本简介.....	1
2. 使用流程.....	2
2.1 流程图.....	2
3. 接口说明.....	3
3.1 结构体说明.....	3
3.1.1 预定义服务.....	3
3.1.2 状态码.....	3
3.1.3 ZUDS_REQUEST	4
3.1.4 ZUDS_RESPONSE	4
3.1.5 ZUDS_FRAME.....	5
3.1.6 ZUDS_SESSION_PARAM	5
3.1.7 ZUDS_ISO15765_PARAM	6
3.1.8 ZUDS_TESTER_PRESENT_PARAM.....	7
3.2 函数说明.....	8
3.2.1 ZUDS_Init.....	8
3.2.2 ZUDS_Request	8
3.2.3 ZUDS_Stop	8
3.2.4 ZUDS_SetTransmitHandler	8
3.2.5 ZUDS_OnReceive.....	8
3.2.6 ZUDS_SetParam.....	9
3.2.7 ZUDS_SetTesterPresent.....	9
3.2.8 ZUDS_Release	9
4. 免责声明.....	10

1. 基本简介

本文档用于说明 UDS 函数库的接口使用方法。函数库的协议层遵循 ISO14229，传输层遵循 ISO15765，支持 ISO15765 的 2004 和 2016 版本，可根据实际应用环境灵活配置。

函数库采用 visual studio 2013 开发，提供的资料有 libudsex.lib、libudsex.dll、zuds.h，同时提供 C++ demo，文档结合 demo，加快二次开发。

函数库事实上并不发送帧数据到总线上，仅通过回调函数把帧数据返回，由用户决定如何发送数据，可采用我司的 zlgcan 函数库进行数据发送，具体使用说明请参考 zlgcan 接口使用手册。

2. 使用流程

2.1 流程图

函数使用流程如图 2.1 所示，请按照图中的流行进行二次开发，具体的接口和函数说明可参考接口说明。

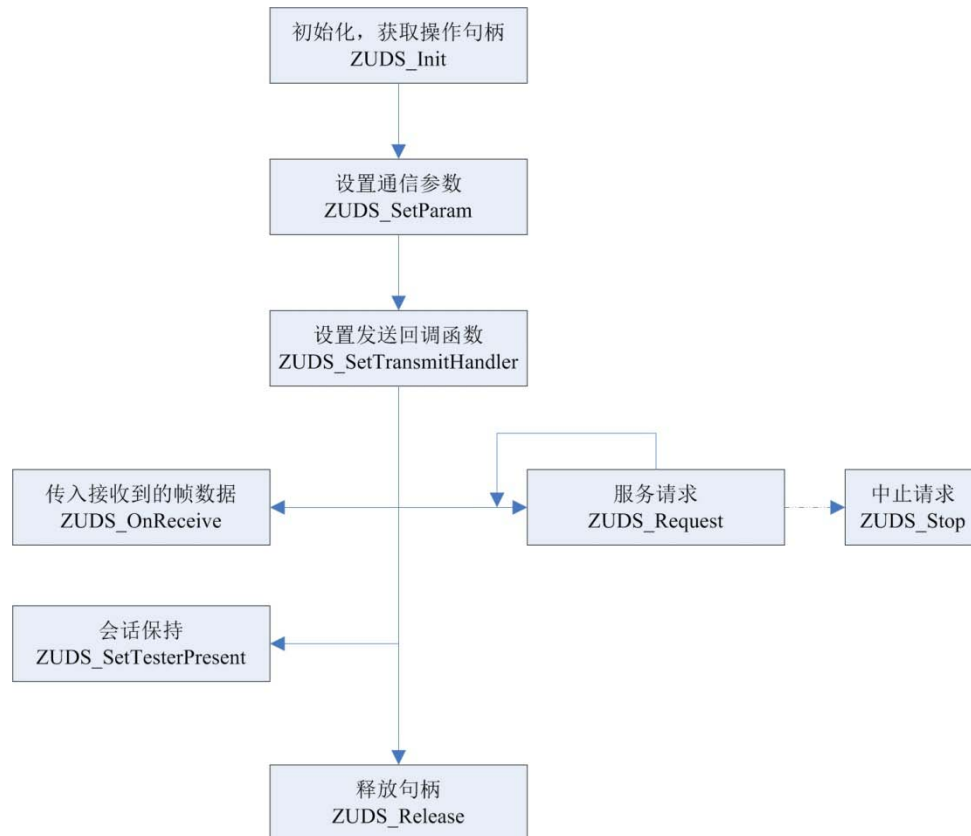


图 2.1 流程图

3. 接口说明

3.1 结构体说明

3.1.1 预定义服务

ISO14229 预定义的服务 ID 如程序清单 3.1 所示，用户亦可添加 ISO 预留用户自定义的服务 ID。

程序清单 3.1 默认服务

#define ZUDS_SI_DiagnosticSessionControl	0x10	// 诊断会话控制
#define ZUDS_SI_ECUReset	0x11	// ECU 重置
#define ZUDS_SI_ClearDiagnosticInformation	0x14	// 清除诊断信息
#define ZUDS_SI_ReadDTCInformation	0x19	// 读取 DTC 信息
#define ZUDS_SI_ReadDataByIdentifier	0x22	// 按标识符读取数据
#define ZUDS_SI_ReadMemoryByAddress	0x23	// 按地址读取内容
#define ZUDS_SI_ReadScalingDataByIdentifier	0x24	// 按标识符读取换算数据
#define ZUDS_SI_SecurityAccess	0x27	// 安全访问
#define ZUDS_SI_CommunicationControl	0x28	// 通讯控制
#define ZUDS_SI_ReadDataByPeriodicIdentifier	0x2A	// 按周期性标识符读取数据
#define ZUDS_SI_DynamicallyDefineDataIdentifier	0x2C	// 动态定义数据标识符
#define ZUDS_SI_WriteDataByIdentifier	0x2E	// 按标识符写数据
#define ZUDS_SI_InputOutputControlByIdentifier	0x2F	// 按标识符的输入输出控制
#define ZUDS_SI_RoutineControl	0x31	// 例程控制
#define ZUDS_SI_RequestDownload	0x34	// 文件下载
#define ZUDS_SI_RequestUpload	0x35	// 请求上传
#define ZUDS_SI_TransferData	0x36	// 数据传输
#define ZUDS_SI_RequestTransferExit	0x37	// 请求传输退出
#define ZUDS_SI_WriteMemoryByAddress	0x3D	// 按地址写内存
#define ZUDS_SI_TesterPresent	0x3E	// 会话保持
#define ZUDS_SI_AccessTimingParameter	0x83	// 访问计时参数
#define ZUDS_SI_SecuredDataTransmission	0x84	// 受保护的数据传输
#define ZUDS_SI_ControlDTCSetting	0x85	// 控制 DTC 设置
#define ZUDS_SI_ResponseOnEvent	0x86	// 基于事件响应
#define ZUDS_SI_LinkControl	0x87	// 链路控制

3.1.2 状态码

程序清单 3.2 列举了响应的状态码，存放于 ZUDS_RESPONSE 的 status 中。

程序清单 3.2 状态码

#define ZUDS_ERROR_OK	0	// no error 无错误
#define ZUDS_ERROR_TIMEOUT	1	// no response until timeout 响应超时
#define ZUDS_ERROR_TRANSPORT	2	// link error 发送帧数据失败
#define ZUDS_ERROR_CANCEL	3	// cancel request 中止请求
#define ZUDS_ERROR_SUPPRESS_RESPONSE	4	// 抑制响应

```
#define ZUDS_ERROR_OTHTER 100 // 其他错误
```

3.1.3 ZUDS_REQUEST

该结构体为请求结构体，如程序清单 3.3 所示，用于存放请求服务的参数，作为 ZUDS_Request 的输入参数。

程序清单 3.3 请求结构体

```
typedef struct _ZUDS_REQUEST
{
    uint32 src_addr;
    uint32 dst_addr;
    uint8  suppress_response;    // 1:suppress
    uint8  sid;                  // service id of request
    uint8  *param;               // array, params of the service
    uint32 param_len;
    uint32 reserved;
}ZUDS_REQUEST;
```

- **src_addr**: 请求地址，即 ECU 地址；
- **dst_addr**: 响应地址，函数库只处理 CAN ID 为响应地址的帧数据；
- **suppress_response**: 是否抑制响应，=1 为抑制响应，即 ECU 没有积极响应返回；
- **sid**: 服务 ID，参考程序清单 3.1；
- **param**: 参数数组，保存服务的参数；
- **param_len**: 参数数组的长度；
- **reserved**: 保留，忽略即可。

3.1.4 ZUDS_RESPONSE

该结构体为响应结构体，如程序清单 3.4 所示，用于存放 ECU 响应的数据，作为 ZUDS_Request 的输出参数。

程序清单 3.4 响应结构体

```
typedef struct _ZUDS_RESPONSE
{
    UDS_STATUS status;
    RESPONSE_TYPE type; // RT_POSITIVE, RT_NEGATIVE
    union
    {
        struct
        {
            uint8  sid;          // service id of response
            uint8  *param;        // array, params of the service, don't free
            uint32 param_len;
        }positive;
        struct
        {

```

```

        uint8  neg_code;           // 0x7F
        uint8  sid;               // service id of response
        uint8  error_code;
    }negative;
};
uint32 reserved;
}ZUDS_RESPONSE;

```

- **status:** 状态码，参考程序清单 3.2，如果状态码不等于 ZUDS_ERROR_OK，不用再解析其他变量；
- **type:** 响应类型，积极响应为 RT_POSITIVE，消极响应为 RT_NEGATIVE；
- **positive.sid:** 积极响应的服务 ID；
- **positive.param:** 积极响应的参数数组，用户不用释放该内存空间，由函数库内部释放；
- **positive.param_len:** 积极响应的参数数组长度；
- **negative.neg_code:** 消极响应的服务 ID，固定为 0x7F；
- **negative.sid:** 消极响应的请求服务 ID；
- **negative.error_code:** 消极响应的错误码；
- **reserved:** 保留，忽略即可。

3.1.5 ZUDS_FRAME

该结构体为帧数据结构体，函数库需要发送的数据将打包成 ZUDS_FRAME 结构回调给用户处理，回调函数请参考 ZUDS_SetTransmitHandler。

用户从外部接收到的帧数据也要转换成 ZUDS_FRAME 结构传给函数库，参考 ZUDS_OnReceive。

程序清单 3.5 帧结构体

```

typedef struct _ZUDS_FRAME
{
    uint32 id;
    uint8  extend;
    uint8  remote;
    uint8  data_len;
    uint8  data[64];
    uint32 reserved;
}ZUDS_FRAME;

```

- **id:** CAN ID；
- **extend:** 是否扩展帧，=1 为扩展帧；
- **remote:** 是否远程帧，=1 为远程帧；
- **data_len:** 帧数据长度；
- **data:** 帧数据数组；
- **reserved:** 保留，忽略即可。

3.1.6 ZUDS_SESSION_PARAM

该结构体用于设置会话层参数，如程序清单 3.6 所示，使用 ZUDS_SetParam 设置给函数库。

程序清单 3.6 会话层参数结构体

```
typedef struct _ZUDS_SERSSION_PARAM
{
    uint16 timeout;           // ms, timeout to wait the response of the server
    uint16 enhanced_timeout; // ms, timeout to wait after negative response: error code 0x78
    uint32 reserved0;
    uint32 reserved1;
}ZUDS_SESSION_PARAM;
```

- **timeout**: 接收响应的超时时间，即 P2 超时；
- **enhanced_timeout**: 接收到消极响应的错误码为 0x78 时，代表 ECU 需要更多的等待时间，即 P2*超时，上位机将继续等待；
- **reserved0**: 保留，忽略即可；
- **reserved1**: 保留，忽略即可。

3.1.7 ZUDS_ISO15765_PARAM

该结构体用于设置基于 ISO15765 实现的传输层的参数，使用 ZUDS_SetParam 设置给函数库。

```
#define VERSION_0 0
#define VERSION_1 1
typedef struct _ZUDS_ISO15765_PARAM
{
    uint8 version;           // VERSION_0, VERSION_1
    uint8 max_data_len;      // max data length, can:8, canfd:64
    uint8 st_min;           // ms, min time between two consecutive frames
    uint8 block_size;
    uint8 fill_byte;        // fill to invalid byte
    uint8 frame_type;       // 0:std 1:ext
    uint32 reserved;
}ZUDS_ISO15765_PARAM;
```

- **version**: 格式版本，= VERSION_0 符合 ISO15765-2 的 2004 版本格式要求，单帧 SF 用第 1 字节低 4 位存放长度，如表 3.1 所示，第一帧 FF 用第 1 字节低 4 位和第 2 字节存放长度，如表 3.2 所示，；= VERSION_1 符合 ISO15765-2 的 2016 版本新增的格式要求，单帧 SF 用第 2 字节存放长度，如表 3.3 所示，第一帧 FF 用第 3~6 字节存放长度，如表 3.4 所示。

表 3.1 格式 1 单帧 SF 格式

字节 1		字节 2~8
高 4 位	低 4 位	
0	长度	负载

表 3.2 格式 1 第一帧 FF 格式

字节 1		字节 2	字节 3~8
高 4 位	低 4 位		
1	长度		负载

表 3.3 格式 2 单帧 SF 格式

字节 1	字节 2	字节 3~N
0	长度	负载

注：N，CAN:8，CANFD：64

表 3.4 格式 2 第一帧 FF 格式

字节 1	字节 2	字节 3~6	字节 7~N
0x10	0	长度	负载

注：N，CAN:8，CANFD：64

- **max_data_len**: CAN 卡只能设置为 8，CANFD 卡如果 version 设置为 VERSION_0 建议设置为 8，VERSION_1 则设置为 64；
- **st_min**: 连续的两个连续帧的最小间隔时间，单位毫秒，存放于上位机发送的流控帧中；
- **block_size**: 允许 ECU 能发送的连续帧的个数，=0 代表后续没有流控帧，即 ECU 可连续把连续帧发送完毕；
- **fill_byte**: 填充字节，无效字节的数值；
- **frame_type**: 帧类型，=1 代表扩展帧；=0 代表标准帧；
- **reserved**: 保留，忽略即可。

3.1.8 ZUDS_TESTER_PRESENT_PARAM

该结构体用于存放会话保持参数，使用 ZUDS_SetTesterPresent 设置给函数库。

```
typedef struct _ZUDS_TESTER_PRESENT_PARAM
{
    uint32 addr;
    uint16 cycle;
    uint8 suppress_response;    // 1:suppress
    uint32 reserved;
}ZUDS_TESTER_PRESENT_PARAM;
```

- **addr**: 会话保持的请求地址；
- **cycle**: 发送周期，单位毫秒；
- **suppress_response**: 是否抑制响应，建议设置为 1；
- **reserved**: 保留，忽略即可。

3.2 函数说明

3.2.1 ZUDS_Init

该函数用于初始化 UDS 函数库，返回操作句柄，用于后续的操作，与 ZUDS_Release 配对使用。

```
typedef uint32 TP_TYPE; // transport protocol
#define DoCAN 0
ZUDS_HANDLE STDCALL ZUDS_Init(TP_TYPE type);
```

- **type**: 传输层协议，目前仅支持 DoCAN;

返回值: 操作句柄，= ZUDS_INVALID_HANDLE 为无效句柄，其他值为有效句柄。

3.2.2 ZUDS_Request

该函数用于服务请求，ECU 有响应即返回，无响应则超时返回，可通过 ZUDS_Stop 中止请求。

```
void STDCALL ZUDS_Request(ZUDS_HANDLE handle, const ZUDS_REQUEST* request,
ZUDS_RESPONSE *response);
```

- **handle**: 操作句柄，为 ZUDS_Init 的返回值，文档接下来的 handle 亦同，不再赘述；
- **request**: 请求结构体，详见 ZUDS_REQUEST；
- **response**: 响应结构体，详见 ZUDS_RESPONSE。

3.2.3 ZUDS_Stop

该函数用于中止请求。

```
void STDCALL ZUDS_Stop(ZUDS_HANDLE handle);
```

3.2.4 ZUDS_SetTransmitHandler

该函数用于设置发送回调函数。函数库自身并不发送帧数据，把打包的帧数据通过回调函数传出给用户发送，用户可通过 zlgcan 函数库进行帧数据发送。

```
#define TRANSPORT_OK 0
#define TRANSPORT_ERROR 1
typedef uint32 (*OnUDSTransmit)(void* ctx, const ZUDS_FRAME* frame, uint32 count);
void STDCALL ZUDS_SetTransmitHandler(ZUDS_HANDLE handle, void* ctx, OnUDSTransmit transmittor);
```

- **OnUDSTransmit**: 回调函数原型；
ctx: 用户设置的参数，原封不动的传出给用户；
frame: 帧数据数组，详见 ZUDS_FRAME；
count: 数组长度；
返回值: 成功则返回 TRANSPORT_OK，否则返回 TRANSPORT_ERROR。

3.2.5 ZUDS_OnReceive

该函数用于把外部接收的帧数据传给函数库，比如用户通过 zlgcan 接收到帧数据，则通过该函数传给 UDS 函数库，否则函数库无法处理响应。

```
void STDCALL ZUDS_OnReceive(ZUDS_HANDLE handle, const ZUDS_FRAME* frame);
```

- **frame**: 详见 ZUDS_FRAME。

3.2.6 ZUDS_SetParam

该函数用于设置函数库的参数。

```
typedef uint8 PARAM_TYPE;  
#define PARAM_TYPE_SESSION    0 // ZUDS_SESSION_PARAM  
#define PARAM_TYPE_ISO15765   1 // ZUDS_ISO15765_PARAM  
void STDCALL ZUDS_SetParam(ZUDS_HANDLE handle, PARAM_TYPE type, void* param);
```

- **type**：参数类型，= PARAM_TYPE_SESSION 用于设置会话层参数，= PARAM_TYPE_ISO15765 用于设置 ISO15765 的通信参数；
- **param**：参数值，type =PARAM_TYPE_SESSION 时为 ZUDS_SESSION_PARAM，type= PARAM_TYPE_ISO15765 时为 ZUDS_ISO15765_PARAM。

3.2.7 ZUDS_SetTesterPresent

该函数用于启动或停止会话保持。

```
void STDCALL ZUDS_SetTesterPresent(ZUDS_HANDLE handle, uint8 enable, const  
ZUDS_TESTER_PRESENT_PARAM* param);
```

- **enable**：=1 代表启动会话保持，如果之前已经启动，则先停止再启动；=0 代表停止会话保持；
- **param**：会话保持参数，详见 ZUDS_TESTER_PRESENT_PARAM。

3.2.8 ZUDS_Release

该函数用于释放资源，与 ZUDS_Init 配对使用。

```
void STDCALL ZUDS_Release(ZUDS_HANDLE handle);
```

4. 免责声明

应用信息

本应用信息中的案例或意图均为假设，仅方便用户熟悉产品的特性以及使用方法。客户在开发产品前必须根据其产品特性给予修改并验证。

修改文档的权利

本手册所陈述的产品文本及相关软件版权均属广州致远电子股份有限公司所有，其产权受国家法律绝对保护，未经本公司授权，其它公司、单位、代理商及个人不得非法使用和拷贝，否则将受到国家法律的严厉制裁。广州致远电子有限公司保留在任何时候修订本用户手册且不需通知的权利。

您若需要我公司产品及相关信息，请及时与我们联系，我们将热情接待。

销售与服务网络

广州致远电子股份有限公司

地址：广州市天河区车陂路黄洲工业区 7 栋 2 楼

邮编：510660

传真：(020)28267891

网址：www.zlg.cn



全国服务电话：400-888-4005

全国销售与服务电话：400-888-4005

销售与服务网络：

广州总公司

广州市天河区车陂路黄洲工业区 7 栋 2 楼

电话：(020)28872342 22644261

上海分公司—上海

上海市北京东路 668 号科技京城东楼 12E 室

电话：(021) 53865521 53083451

北京分公司

北京市海淀区知春路 108 号豪景大厦 A 座 19 层

电话：(010)62536178 62635573

上海分公司—南京

南京市珠江路 280 号珠江大厦 1501 室

电话：(025) 68123923 68123920

深圳分公司

深圳市福田区深南中路 2072 号电子大厦 12 楼

电话：(0755)83640169 83783155

上海分公司—杭州

杭州市天目山路 217 号江南电子大厦 502 室

电话：(0571)89719491 89719493

武汉分公司

武汉市洪山区广埠屯珞瑜路 158 号 12128 室(华中电脑数码市场)

电话：(027) 87168497 87168397

重庆分公司

重庆市九龙坡区石桥铺科园一路二号大西洋国际大厦(赛格电子市场) 2705 室

电话：(023)68796438 68797619

成都分公司

成都市一环路南二段 1 号数码科技大厦 403 室

电话：(028) 85439836 85432683

西安办事处

西安市长安北路 54 号太平洋大厦 1201 室

电话：(029)87881295 87881296