



六叶树 DBC 解析库 API 使用手册

文档信息

项目	信息
产品名称	六叶树 CAN 总线 DBC 解析库 (lysdbc)
当前版本	V1.0.0
适用平台	Windows / Linux 跨平台
适用场景	车载 CAN 开发、ECU 诊断、总线监控、自动化测试
开发公司	六叶树教育科技有限公司

1 文档概述

1.1 库简介

六叶树 CAN 总线 DBC 解析库 (lysdbc) 为自主研发的轻量化、高稳定性车载 CAN 协议解析开发库。库内封装 DBC 文件解析、CAN/CAN FD 报文编解码、信号物理值换算、总线协议适配等底层功能。接口简洁通用，无第三方依赖，适配工业级车载开发标准，广泛应用于整车总线调试、ECU 功能测试、车载诊断上位机开发等场景。

1.2 核心功能特性

- **报文兼容：**全面支持标准 CAN、CAN FD 报文解析与编码
- **文件加载：**支持单 DBC 覆盖加载、多 DBC 合并叠加加载
- **信号换算：**原生支持原始值、物理值双向高精度换算



- 枚举解析：内置信号文本描述映射，适配故障码、状态位解析
- 协议适配：支持 J1939 商用车协议、自定义私有 CAN 协议
- 动态管理：运行时动态增、删、改、查 CAN 报文
- 跨平台：无缝兼容 Windows、Linux 系统

2 宏定义与枚举常量

2.1 通用错误码

宏定义	数值	详细说明
ERR_SUCCESS	0	接口执行成功
ERR_FAILED	1	接口执行失败
ERR_MULTI_TRANSMITTING	2	多帧传输正在进行
ERR_NOT_SUPPORT	3	当前功能暂不支持
INVALID_DBC_HANDLE	0	无效 DBC 操作句柄

2.2 帧类型定义

宏定义	数值	详细说明
OBJ_CAN	0	标准 CAN 报文（8 字节）
OBJ_FD	1	CAN FD 报文（最长 64 字节）



2.3 总线协议类型

宏定义	数值	详细说明
PROTOCOL_J1939	0	J1939 商用车专用协议
PROTOCOL_OTHER	1	自定义私有 CAN 协议

3 核心数据结构体

本章详细说明库内核心结构体，为编解码、信号换算、文件加载的数据基础，所有结构体遵循 C 语言标准，可直接二次开发。

3.1 ValDescPair 信号枚举描述结构体

用于存储信号数值与文本描述映射关系，适用于档位、开关、故障状态等枚举类信号。

```
c
typedef struct
{
    double value;           // 信号原始数值
    char desc[DBC_VAL_DESC_LEN+1]; // 数值对应文字描述
}ValDescPair;
```

3.2 DBCSignal 信号结构体

存储单路 CAN 信号全部解析参数，包含位偏移、换算系数、字节序、量程、复用规则等。

```
c
typedef struct
{
    uint32 nStartBit;       // 信号起始位
    uint32 nLen;           // 信号位长度
```



```
double nFactor;           // 比例换算系数
double nOffset;          // 物理值偏移量
double nMin;              // 信号最小值
double nMax;              // 信号最大值
uint64 nRawValue;        // 总线原始数值
bool isSigned;            // 是否为有符号数据
bool isMotorola;         // 是否为摩托罗拉字节序
uint8 multiplexerType;   // 信号复用类型
uint8 valueType;         // 数据存储类型
uint32 multiplexerValue; // 复用匹配判定值
char unit[DBC_UNIT_MAX_LENGTH+1]; // 物理单位
char strName[DBC_NAME_LENGTH+1]; // 信号名称
char strComment[DBC_COMMENT_MAX_LENGTH+1]; // 注释
}DBCSignal;
```

3.3 DBCMessage 报文结构体

一条 CAN 报文完整数据载体，包含报文属性、信号数组、时序参数。

```
c
typedef struct
{
    uint32 nSignalCount;           // 报文内信号数量
    uint32 nID;                    // CAN 报文 ID
    uint32 nSize;                  // 报文数据长度
    double nCycleTime;             // 报文发送周期
    uint8 nExtend;                 // 0=标准帧 1=扩展帧
    DBCSignal vSignals[DBC_SIGNAL_MAX_COUNT]; // 信号数组
    char strName[DBC_NAME_LENGTH+1]; // 报文名称
    char strComment[DBC_COMMENT_MAX_LENGTH+1]; // 报文注释
}DBCMessage;
```

3.4 FileInfo DBC 加载配置结构体



用于 DBC 文件加载参数配置，控制文件路径与叠加加载逻辑。

```
c
struct FileInfo
{
    char strFilePath[DBC_MAX_FILE_PATH+1]; // DBC 文件路径
    uint8 merge;           // 0=清空覆盖 1=合并叠加加载
};
```

4 API 函数详细手册

本章按照调用逻辑排序，包含函数功能、函数原型、入参、返回值、使用说明，为开发标准调用规范。

4.1 库初始化与资源释放

4.1.1 LDBC_Init —— 创建 DBC 操作句柄

功能描述：初始化 DBC 解析内存池，创建唯一操作句柄，所有 API 调用必须依赖该句柄。

函数原型：DBCHandle LDBC_Init(void);

返回值：非 0=有效句柄；0=初始化失败。

4.1.2 LDBC_Release —— 释放库资源

功能描述：销毁句柄、释放全部堆内存，防止内存泄漏，程序结束必须调用。

函数原型：void LDBC_Release(DBCHandle hDBC);

入参：hDBC —— 初始化生成的 DBC 句柄。

4.2 DBC 文件加载接口

4.2.1 LDBC_LoadFile —— 加载 DBC 协议文件



功能描述: 解析本地 DBC 文件，导入报文、信号、换算参数，支持多文件合并。

函数原型: `bool LDBC_LoadFile(DBCHandle hDBC, const FileInfo* pFileInfo);`

注意事项: 中文文件路径必须使用 GB2312 编码，UTF-8 中文路径无法识别。

4.3 总线协议配置接口

4.3.1 LDBC_SetProtocol —— 设置总线协议

原型: `void LDBC_SetProtocol(DBCHandle hDBC, int type);`

说明: 选择 J1939 协议或自定义 CAN 协议，适配不同车型总线规范。

4.3.2 LDBC_GetProtocol —— 获取当前协议

原型: `uint8 LDBC_GetProtocol(DBCHandle hDBC);`

4.4 报文信息读取接口

4.4.1 LDBC_GetMessageCount

获取当前已加载全部 CAN 报文总数量。

4.4.2 LDBC_GetMessageByIndex

通过索引顺序，读取单条报文完整结构体信息。

4.5 信号枚举描述接口

4.5.1 LDBC_GetValDescPairCount

获取指定信号的枚举描述条数。

4.5.2 LDBC_GetValDescPair



读取信号全部枚举映射表，用于上位机状态文字展示。

4.6 报文解析接口（总线数据解码）

4.6.1 LDBC_Decode

核心接口：输入原始 CAN/CAN FD 帧，自动按 DBC 规则解析为物理工程值。

函数原型：`bool LDBC_Decode(DBCHandle hDBC, const void* pObj, uint8 objType, DBCMessage* pMsg);`

4.7 报文编码接口（工程值编码总线）

4.7.1 LDBC_EncodeToObj

将 DBC 报文结构体编码为 CAN 帧对象，为最常用发送接口。

4.7.2 LDBC_EncodeToBuffer

将报文编码为纯二进制字节数组，适配底层硬件发送。

4.8 信号换算工具接口

4.8.1 LDBC_CalcPhysicalValue

换算公式：物理值 = 原始值 × factor + offset

4.8.2 LDBC_CalcRawValue

换算公式：原始值 = (物理值 - offset) / factor

4.9 动态报文管理接口

- **LDBC_ClrMessage:** 清空内存中所有已加载报文



- **LDBC_InsertMessage:** 手动自定义插入/覆盖 CAN 报文
- **LDBC_DeleteMessage:** 根据报文 ID 删除指定报文

5 工程标准示例代码

本段代码为工业级标准 Demo，可直接编译运行，适配所有开发环境。

```
c
#include <stdio.h>
#include <string.h>
#include "lysdbc.h"

int main()
{
    // 1. 初始化 DBC 解析库
    DBCHandle hDbc = LDBC_Init();
    if (hDbc == INVALID_DBC_HANDLE)
    {
        printf("DBC 初始化失败\n");
        return -1;
    }

    // 2. 配置并加载 DBC 文件
    FileInfo fileInfo = {0};
    strcpy(fileInfo.strFilePath, "car.dbc");
    fileInfo.merge = 0;           // 覆盖加载模式

    if(!LDBC_LoadFile(hDbc, &fileInfo))
    {
        printf("DBC 文件加载失败\n");
        LDBC_Release(hDbc);
        return -1;
    }
}
```


```
// 3. 设置总线协议类型
LDBC_SetProtocol(hDbc,PROTOCOL_OTHER);

// 4. 模拟硬件接收一帧 CAN 报文
CanObj canFrame = {0};
canFrame.ID = 0x123;
canFrame.DataLen = 8;

// 5. 解析报文并打印信号
DBCMessage outMsg = {0};
if(LDBC_Decode(hDbc,&canFrame,OBJ_CAN,&outMsg))
{
    for(uint32 i=0;i<outMsg.nSignalCount;i++)
    {
        double val =
LDBC_CalcPhysicalValue(&outMsg.vSignals[i],outMsg.vSignals[i].nRaw
Value);
        printf("信号:%s
值:%.2f\n",outMsg.vSignals[i].strName,val);
    }
}

// 6. 销毁句柄、释放内存
LDBC_Release(hDbc);
return 0;
}
```

6.联系我们

 电话:15211065817(业务合作咨询)

 邮箱:798746621@qq.com(业务咨询+技术支持)



六叶树 DBC 库接口函数使用手册



微信:18673379565(技术支持)



官网:www.liuyeshu.cn(资料下载)



网上商城:<https://shop112408209.taobao.com>(产品购买)

淘宝店铺搜索:“六叶树教育科技”