

# USB-CAN A1 A2 dll 接口函数包使用手册

## DLL 版本 V2.0

### 1 概述

本手册是操作 USB-CAN A2 应用函数的定义格式、调用方法和返回值的说明。

#### 1.1 硬件设备 **USB-CAN A2**

1.2 使用前可使用 demo 带的测试函数试验。

1.3 Qm\_usb.dll 是接口 DLL;

1.4 Qm\_usb.dll 必须放置到应用软件 Exe 文件所在目录下。

1.5 需要 Windows XP、Windows 2000 或以后版本

### 2 函数说明

#### 2.1 初始化设备

##### 2.1.1 函数:

Init\_can(com\_NUM,Model,CanBaudRate,SET\_ID\_TYPE,FILTER\_MODE,RXF,RXM)

2.1.2 说明: 本函数用于初始化 USB 接口, 并对 CAN 总线进行基础设置。

##### 2.1.3 参数:

CanBaudRate: CAN 波特率

值	意义	值	意义
5	5 KBPS	125	125 KBPS
10	10 KBPS	200	200 KBPS
15	15 KBPS	250	250 KBPS
20	20 KBPS	400	400 KBPS
25	25 KBPS	500	500 KBPS
40	40 KBPS	667	666.7 KBPS
50	50 KBPS	800	800 KBPS
80	80 KBPS	1000	1M BPS
100	100 KBPS		

注意: 只有上表列出的值有效, 无他数值均无效。如果设置成其他数值, 将导致设置失败。

com\_NUM: 设备连接的端口。0: 自动选择, 其他值: 端口号

Model: 设备类型 1: A1 型 2: A2 型 3: A3 型

SET\_ID\_TYPE: 设置 ID 的排列方式 各种 CAN 芯片的 ID 排列是不同的, 设置此项以后, 可以使收发 ID 都按照客户习惯的方式排列

值	意义
0	2515 方式
1	SJA1000 方式
2	右对齐方式

FILTER\_MODE: 设置接收过滤模式

值	意义
0	不开启过滤功能, 接收全部报文

1	只接收符合条件的标准帧
2	只接收符合条件的扩展帧

**RXF:** 滤波寄存器 四字节数组。无论是否使用此功能，均需声明四字节数组，过滤标准帧时前两字节有效

**RXM:** 屏蔽寄存器 四字节数组。无论是否使用此功能，均需声明四字节数组，过滤标准帧时前两字节有效

函数返回值：0：成功 其他值：错误

#### 2.1.4 声明与调用示例

Function

Init\_can(com\_NUM:BYTE;Model:BYTE;CanBaudRate:WORD;SET\_ID\_TYPE:BYTE;FILTER\_MODE:BYTE;RXF:pointer;RXM:pointer):integer;stdcall external 'QM\_USB.dll';

```
var ret:integer;
    RXF,RXM:array [0..3] of BYTE;
    ret:=Init_can(0,2,20,0,2,@RXF,@RXM); //初始化函数演示
```

#### 2.2 发送数据

2.2.1 函数：Can\_send(IDbuff:,Databuff:,FreamType,Bytes)

2.2.2 说明：本函数用于发送 CAN 数据。

2.2.3 参数：

IDbuff: ID 缓冲区 四字节数组。

Databuff: 数据缓冲区 八字节数组。

FreamType: 发送的帧类型。

值	意义
0	标准数据帧
1	扩展数据帧
2	标准远程帧
3	扩展远程帧

Bytes: 发送的有效数据字节个数

函数返回值：0：成功 其他值：错误

#### 2.2.4 声明与调用示例：

Function Can\_send(IDbuff:pointer;Databuff:pointer;FreamType:Byte;Bytes:Byte):integer;stdcall external 'QM\_USB.dll';

Var

IDbuff:array[0..3] of byte;

Databuff:array [0..7] of byte;

```
ret:= Can_send(@IDbuff,@Databuff,0,8); //发送函数演示
```

#### 2.3 接收数据

2.3.1 函数：Can\_receive(IDbuff,Databuff,FreamType,Bytes)

2.3.2 说明：本函数用于查询接收缓冲区，无数据返回 0，有数据返回 1，并接收数据。

### 2.3.3 参 数:

IDbuff: ID 缓冲区 四字节数组。

Databuff: 数据缓冲区 八字节数组。

FreamType: 收到的帧类型。

值	意义
0	标准数据帧
1	扩展数据帧
2	标准远程帧
3	扩展远程帧

Bytes: 收到的数据字节个数

函数返回值: 0: 无数据, 1: 收到数据, 其他值: 错误

### 2.3.4 声明与调用示例:

function

Can\_receive(IDbuff:pointer;Databuff:pointer;FreamType:pointer;Bytes:pointer):integer;stdcall  
external 'QM\_USB.dll';

var

ret,i:integer;

IDbuff:array[0..3] of byte;

Databuff:array [0..100] of byte;

FreamType,Bytes:byte;

ret:=Can\_receive(@IDbuff,@Databuff,@FreamType,@Bytes);//接收函数演示

//使用时需不断调用。建议放在定时器里。

## 2.4 接收数据

2.4.1 函 数: Quit\_can();

2.4.2 说 明: 本函数用于查询接收缓冲区, 无数据返回 0, 有数据返回 1, 并接收数据。

2.4.3 函数返回值: 0: 成功 其他值: 错误

2.4.4 声明与调用示例: Quit\_can(); //释放函数演示。退出程序或需重设参数需设用此函数。

## 3 一个完整的示例

参考 examples 文件夹下, 程序例子。